UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Tècnica Superior d'Enginyeria
de Telecomunicació de Barcelona

ZERO SPEED ROTOR POSITION ESTIMATOR BASED ON SLIDING MODE CONTROL FOR PERMANENT MAGNET SYNCHRONOUS MOTOR

A Master's Thesis

Submitted to the Faculty of the

Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona

Universitat Politècnica de Catalunya

by

Jabir Bin Waqar

In partial fulfilment

of the requirements for the degree of

MASTER IN ELECTRONICS ENGINEERING

Advisors: Domingo Biel Solé; Víctor Repecho Del Corral

Barcelona, February 2020

Title of the thesis: ZERO SPEED ROTOR POSITION ESTIMATOR BASED ON SLIDING MODE CONTROL FOR PERMANENT MAGNET SYNCHRONOUS MOTOR

Author: Jabir Bin Waqar

Advisors: Domingo Biel Solé; Víctor Repecho Del Corral

Abstract

Implementation of an algorithm based on SMC that uses a unique technique for the rotor position estimation of a PMSM for low and zero speeds, by using inherit motor effect called “Saliency”. The work proves that the rotor position estimation is possible with the information that is present in the system under SMC due to the saliency effect. Therefore, there is no need to inject any signal into the machine, which causes the increment in the losses of the machine, or to design dynamic observers. The algorithm is implemented in a DSP controller and the tests with the complete hardware platform validate the proposal in open loop and in sensorless operation.

Acknowledgements

Foremost, I would like to express my sincere gratitude to my advisor Prof. Domingo Biel Solé for the continuous support of my Master thesis and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis.

Besides my advisor, I would really like to thank my co-advisor Prof. Víctor Repecho Del Corral for his continuous support, encouragement, insightful comments, experimentation and data analysis and to provide phenomenal technical insights to work with the real time systems.

Finally, I cannot express enough thanks to my Master advisor Prof. Isidro Martin Garcia and my other professors and advisors for their continued support and encouragement during the whole period of Master study. I offer my sincere appreciation for the learning opportunities provided by my professors throughout the entire time.

Revision history and approval record

Revision	Date	Purpose
0	10/01/2020	Document creation
1	26/01/2020	Document revision 1
2	26/02/2020	Document revision 2
3	04/03/2020	Document final revision

Written by:		Reviewed and approved by:	
Date	10/01/2020	Date	04/03/2020
Name	Jabir Bin Waqar	Name	Domingo Biel Solé; Víctor Repecho Del Corral
Position	Project Author	Position	Project Supervisor

Table of contents

List of Figures

List of Tables

1. Introduction

The goal of this study is directly related with the field of power electronics, particularly with power control for renewable energy sources that uses a number of advance control techniques for power systems. Some of those control techniques are: Fuzzy control, Lyapunov based control and Sliding Mode Control (SMC), etc.

This work is mainly focused on permanent magnet synchronous motor, but before introducing the permanent magnet synchronous motor (PMSM) or any other parts of the work, a short introduction to the machines is provided in the following section.

1.1. Asynchronous and synchronous machines

In the past, or even until today, the machines that are mostly used in the industry are induction machines or also known as asynchronous machines. This is due to that fact that induction machines are simple to manufacture, they are cheap, and they do not require a complex control system, unlike the other synchronous machines or any type of dc machines. Additionally, an induction motor does not have brushes into it which drastically reduce the maintenance of the motor.

Figure (1-1) shows a simplified version of the basic structure of an induction motor in which the stator coils and a squirrel cage rotor can be observed.

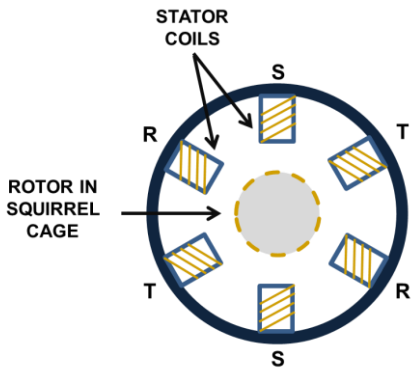


Figure 1-1: Simplified structure of an induction motor

In general, each motor has its own advantages and disadvantages, e.g., dc motors have high initial cost and also high maintenance cost due to presence of commutator and brush gears and they can be used for low to medium power. To go to higher powers, the motors that are commonly used are induction/asynchronous motors as described before. One of the main advantages of these motors is that the control does not require the rotor angle position to move the motor, which simplifies the complexity of the control. The synchronous motors can improve the performance over the induction ones, but the control complexity also increases. Moreover, among the synchronous machines, the permanent magnet ones have the highest power density.

As this work is only focused on permanent magnet synchronous motor, making it the main part of the real system on which sliding mode control is applied and from there rotor angle is estimated, a detailed explanation including: types, construction and modelling equations of a permanent magnet synchronous motor are described in the following sections of this Chapter.

1.2. Permanent magnet synchronous motor (PMSM)

In recent times, the new design techniques of magnets have made it possible to design and expand the field of some known motors like permanent magnet synchronous motor. These types of motors have the peculiarity of owning a non-wounded rotor, instead using a permanent magnet, where the magnetic field is constant, and it does not require a motor exciter for the rotor. This characteristic of the motor provides several advantages like a good power to volume ratio, better efficiency and reliability as compared to an induction motor. Nevertheless, these types of machines are much more complex when it comes to control as compared to induction motors, because of the need to use an encoder/sensor to know the rotor position all the time.

Figure (1-2) shows a simplified version of the basic structure of a permanent magnet synchronous motor in which the stator coils and a movable rotary magnet can be observed.

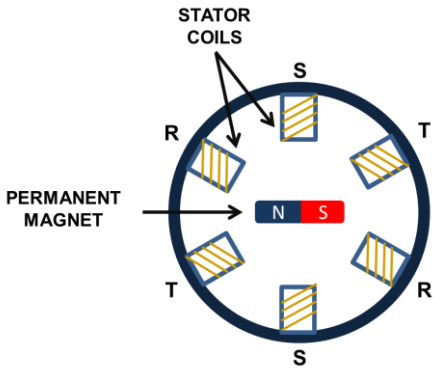


Figure 1-2: Structure of a permanent magnet synchronous motor (PMSM)

There are two main types of permanent magnet synchronous motor:

1. Interior permanent magnet synchronous motor (IPMSM)
2. Surface permanent magnet synchronous motor (SPMSM)

Figure (1-3) represents IPMSM and SPMSM in subfigures (a) and (b), respectively.

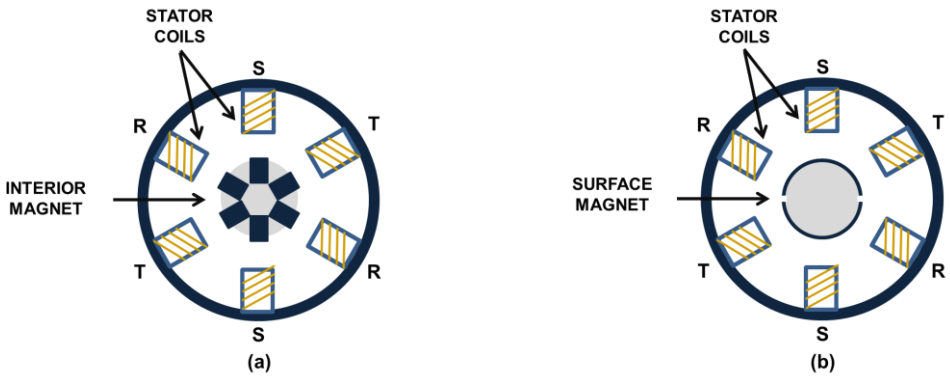


Figure 1-3: Main constructional types of PMSM

1.2.1. Construction of PMSM

The construction of this motor in a very simple form can be seen in the Figure (1-2). In the structure of the stator, the placement of the coils is very similar to an induction or synchronous motor presented in the Figure (1-1), but in the construction of its rotor it

becomes totally different from its predecessors. As the name permanent magnet synchronous motor indicates that the rotor is made of a permanent magnet that generates the magnetic flux. In an induction motor this flux is called induced flux and into other synchronous motors this flux is provided externally, but in PMSM the rotor itself magnetizes and generates the flux.

Lately, there has been a lot of growth of interest for these motors, and one of the main reasons for that is to find new materials to manufacture the magnets, and definitely, there has been done a lot of improvement in the properties of these motors with the new materials. Without getting into details, the materials that are most commonly being used to manufacture the magnets are: Samarium–cobalt (SmCo) and Neodymium iron boron (NdFeB) having the last one with great magnetizing power.

1.2.2. Modelling of PMSM

After introducing a short description about the technologies and functioning of the commonly used motors, as all this work is based on PMSM alone, now it is time to proceed and introduce the mathematical/electrical model/s of the permanent magnet synchronous motor.

The electrical model of PMSM with saliency in abc frame is derived as following:

$$L\frac{di}{dt} = v_{abcn} - \left(RI + \frac{dL}{dt}\right)i - e(\omega_e, \theta_e) \quad (1.1)$$

Where;

$i = [i_a \ i_b \ i_c]^T$ is a vector of stator currents. $e(\omega_e,\theta_e)$ contains the values of back electromagnetic forces (back-EMFs), where ω_e and θ_e are the speed and rotor angle position respectively. R represents the stator winding resistance, I is a $3x3$ identity matrix And $v_{abcn} = [v_{an} \ v_{bn} \ v_{cn}]^T$ is a vector that represents stator voltages with respect to point n that can be observed in Figure (1-6) as vn.

Now assuming that the stator windings are distributed uniformly along the motor, which gives the inductance matrix L that is the following:

$$L = \begin{bmatrix} L_a & L_{ab} & L_{ac} \\ L_{ab} & L_b & L_{bc} \\ L_{ac} & L_{bc} & L_c \end{bmatrix} \quad (1.2)$$

Where L_a, L_b and L_c are the self-inductances of stator windings and L_{ab}, L_{bc} and L_{ac} are the mutual inductances that correspond to the following:

$$L_a = L_m + \Delta L\cos(2\theta_e) \quad (1.3)$$

$$L_b = L_m + \Delta L\cos\left(2\theta_e + \frac{2\pi}{3}\right) \quad (1.4)$$

$$L_c = L_m + \Delta L\cos\left(2\theta_e - \frac{2\pi}{3}\right) \quad (1.5)$$

$$L_{ab} = -\frac{L_m}{2} + \Delta L\cos\left(2\theta_e - \frac{2\pi}{3}\right) \quad (1.6)$$

$$L_{ac} = -\frac{L_m}{2} + \Delta L\cos\left(2\theta_e + \frac{2\pi}{3}\right) \quad (1.7)$$

$$L_{bc} = -\frac{L_m}{2} + \Delta L\cos(2\theta_e) \quad (1.8)$$

Where L_m and ΔL are motor parameters, which are related to well-known L_d and L_q inductances in the dq frame (where dq refers to direct and quadrature components). The term ΔL is known as motor saliency, and from the control point of view it is often neglected since $L_m >> \Delta L$.

Finally, considering the uniform distribution of the stator windings, the voltage at point n of the motor can be approximated by:

$$v_n = \frac{v_{bus}}{3}[u_a + u_b + u_c] \quad \textbf{(1.9)}$$

Remark I: *dq frame is another very famous mathematical model of the PMSM that is obtained by applying following transformations:*

1. *First apply* ***Clarke*** *transformation on the real abc system (this simplifies the complex equations and changes the axes of the abc system to alfa-beta-gamma domain, and if the system in abc is well-balanced then gamma becomes zero and new system reduces to two orthogonal axes alfa-beta only.*
2. *Then apply* ***Park*** *transformation on alfa-beta domain (which synchronizes the system by aligning the alfa-beta axes with the rotor, hence changing the domain to dq domain. Thus, new model reduces to dq frame only).*

More details about these transformations and/or reverse transformations can be found in the reference [1].

1.3. Control techniques applied to PMSM

Once the mathematical model of the PMSM is introduced, now it is time to introduce the different control techniques to control the motor.

Some control techniques that are commonly used are described as following [2]:

1. Linear control techniques
    a. PI-Controller in Field Oriented Control (FOC)
    b. Direct Torque Control (DTC) with Space Vector Modulation (SVM)
2. Non-linear control techniques
    a. Non-linear Torque Control
    b. Hysteresis Current Control
    c. Fuzzy Logic Control
    d. Sliding Mode Control (SMC)

Remember that, in order to perform any of these controls for PMSM, a variable that must be known all the time to control the motor properly is the rotor angle position.

1.3.1. Examples

To elaborate a little bit more about the mentioned techniques, two control system examples have been provided.

1.3.1.1. PI-Controller in Field Oriented Control (FOC)

A FOC controller is also called a vector controller that uses the dq model of the motor to implement linear (PI) controllers to control the currents of the system. The current i_d is referenced to zero in order to avoid the demagnetization of the magnet, and i_q* can be generated by using a speed controller or a torque reference. Hence by controlling i_d and i_q currents, motor can be controlled with a known rotor position [3].

Figure (1-4) depicts a block diagram of a FOC control implementation for dq model.

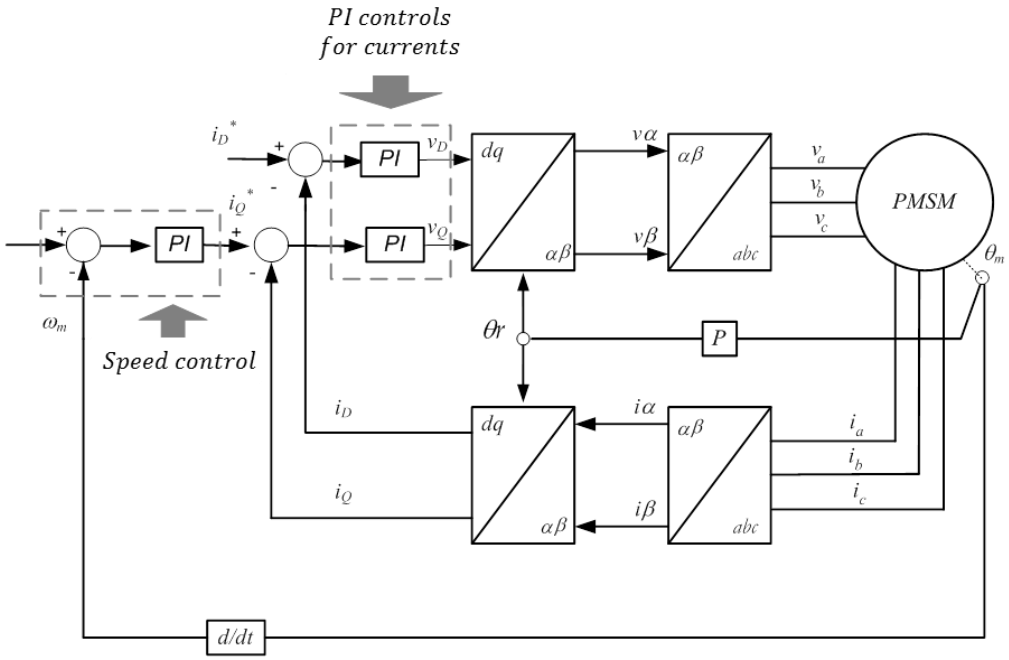


Figure 1-4: FOC implementation in dq model for PMSM

As we know a PI controller is related with the working point obtained by using a linearized small signal model of the system, therefore it becomes a drawback of the PI controller.

1.3.1.2. Sliding Mode Control (SMC)

A SMC is a non-linear controller that can also be implemented in dq model. The current i_d is referenced to a zero value like in FOC control and i_q* can be generated by using a speed controller or a torque reference. So, by using hysteresis comparators, this control can also be implemented, considering that the rotor position is always known, but the implementation of this control in dq model is very complex [3].

Since the real system is a three-phase system, normally known as abc system and the controls that are derived from a dq model are too complex and cannot be directly applied to abc system. Thus, SMC is designed in abc system for the easy implementation of the controls.

Figure (1-5) depicts a block diagram of a SMC control implementation for dq model and Figure (1-6) describes a block diagram of a SMC control applied to an abc system.

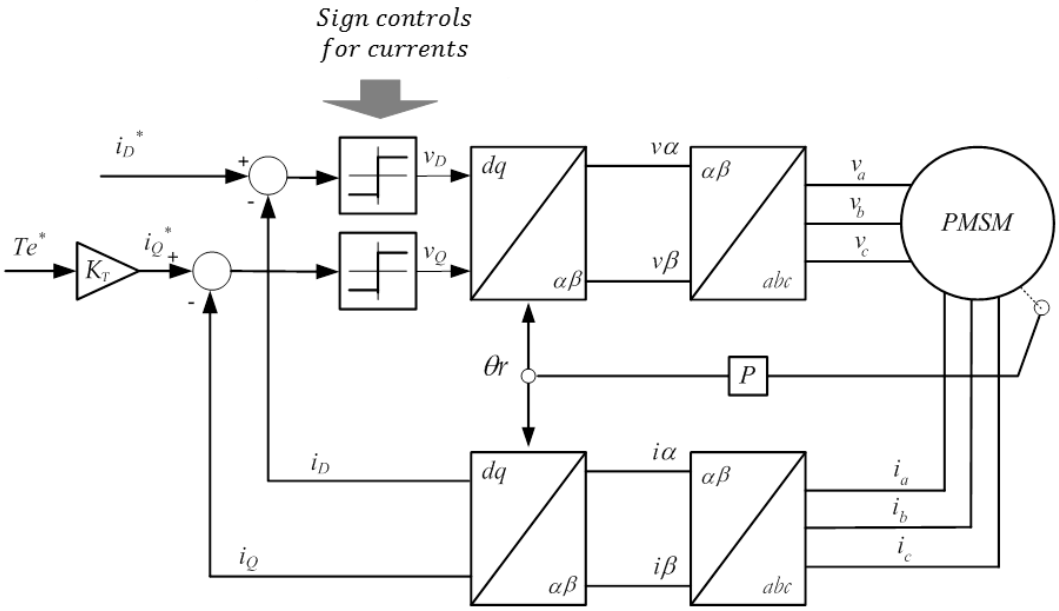


Figure 1-5: SMC implementation in dq model for PMSM

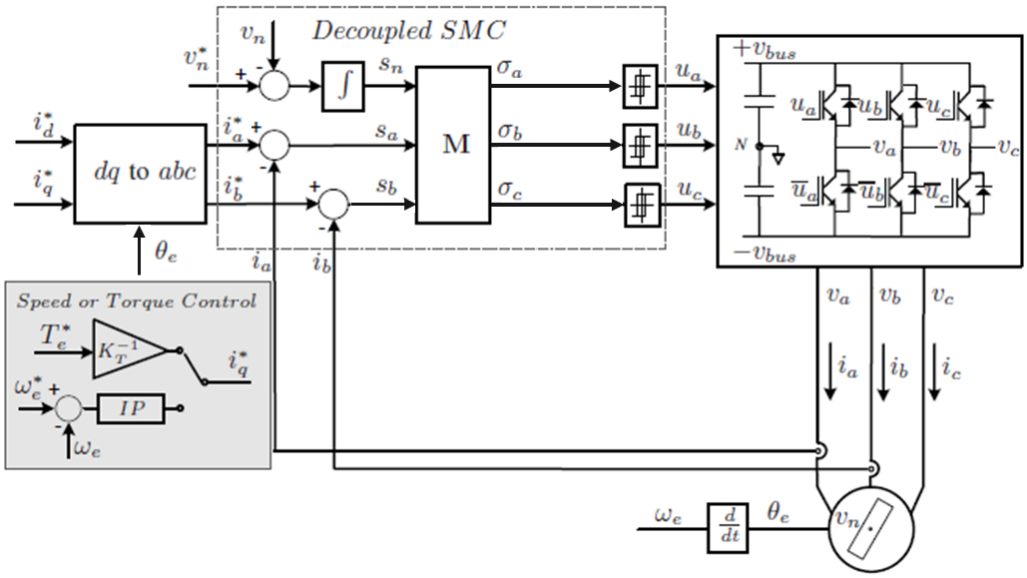


Figure 1-6: SMC implementation in abc model for PMSM

Decoupled sliding mode control is one of the several methods that can be applied to an abc system. Furthermore, in this work this technique has been used. The design and implementation of this control method can be found in Chapter 3 “Design of zero speed rotor position estimator based on SMC for PMSM”, in section “Decoupled SMC applied to PMSM”.

Remark II: *Unlike a linear controller such as PI, a SMC is derived from the large signal model of the system thus it uses full dynamics of the system. Therefore, SMC controller has much more dynamical range as compared to a linear controller obtained with linearized small signal model of the system.*

1.4. Need to know the rotor position

As it is stated already, a known rotor angle position is very important to perform any kind of control on such motors. Normally the rotor position is tracked by using an encoder which increases the complexity of control, can be very expensive and increases the overall cost of the system; due to this fact it has taken a lot of attention in the past years. The idea that has been chased is to somehow remove the need of encoder from the system and get the rotor angle position signal by other means. Additionally, the estimator can be also used as a secondary rotor sensor, means providing a backup in case if the primary one fails.

A lot of research work has been done on this topic and people have been trying to implement new methods to either estimate the rotor angle position or to design some observers to track the rotor position without using any kind of decoder and to make the system simple and cheap. Most of the existing algorithms estimated the angle only if a motor is rotating and motor speed is above a specific limit. This implies that once the motor speed is reduced below the limits, the algorithms stops working, thus there is no more information regarding the rotor position for low and zero speeds.

Being said that, there are some works in the literature where people have been succeeded to estimate the rotor angle position for low and zero speeds, but they are using some techniques that are very complex, require high performance system for the execution, increase the losses in the motor and decrease the overall efficiency of the system [4]. More detailed information regarding those works has been provided in the following Chapter 2 “State of the art of estimators and observers on PMSM”.

Hence to conclude, this work is a huge contribution into the estimator’s part of these technologies with considerable advantages over all the existing estimators or even on the observers that are being used for angle estimation.

1.5. Proposal of this thesis

The main goal is to design an algorithm to estimate rotor angle, which is simple, does not require any signal injection and it works for low and zero speeds.

The implementation of this algorithm is in such a way that it extracts the information about the rotor position from the inherit effect of the motor, called “Saliency”, under SMC.

1.5.1. Objectives

- Design and implementation of a digital sliding mode control and rotor position estimation algorithm
- Test the rotor position estimation algorithm in a DSP system and obtain good results at zero speed

1.6. Work plan with tasks, milestones and a Gantt chart

A brief description of this work is presented in this topic with a general work plan described in following subtopic “Work plan and milestones” and then initial planning, a more detailed version of the tasks related with time for this work are presented in the subtopic “Gantt diagram”.

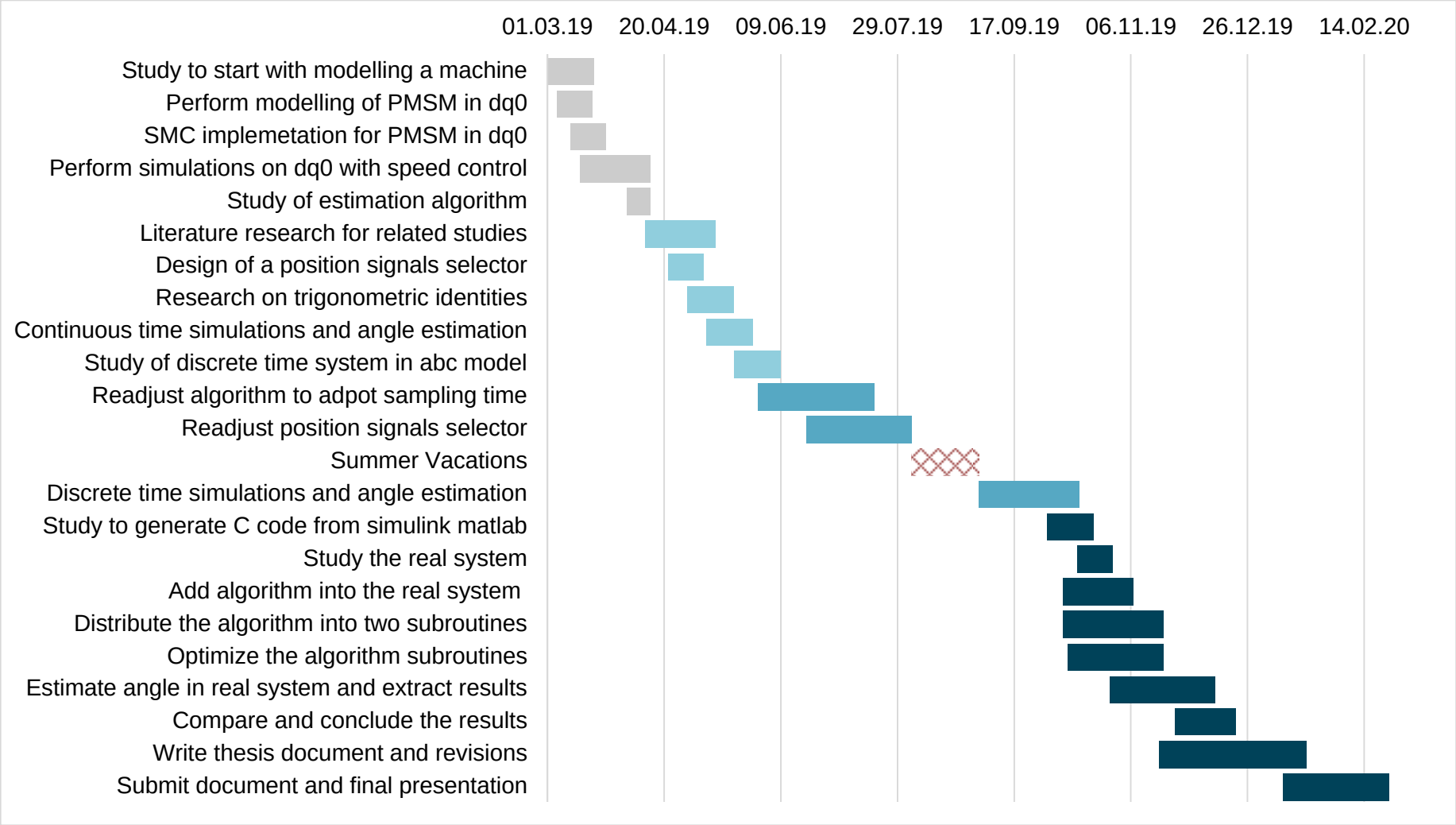
1.6.1. Work plan and milestones

Followings are the generalization of the main tasks proposed for this work:

- Literature study for similar technologies
- Introduction to SMC on PMSM
- Study of the proposed algorithm
- Perform ideal simulations and get angle estimation
- Perform discrete time simulations and get angle estimation
- Execute the algorithm in a DSP system
- Compare discrete time simulation results with the real system results
- Conclusions of the comparison of the results
- Future developments

1.6.2. Gantt chart

Table 1: Gantt chart



2. State of the art of estimators and observers on PMSM

Due to increasingly used of PMSM in high performance applications, a significant attention has been focused on designing the control techniques for this machine which require the rotor position to perform a proper control as it is said before. There are research groups and individuals that are emerging in the race to develop a fully sensorless stand-still control with no need of an encoder/sensor for the motor at all [5]. A lot of research work has been done and still in progress which verifies the idea of how interesting this topic has become.

To cover all the technological ground in this field, the state of the art of these technologies can be divided into two main sections:

1. Rotor position estimation (RPE)
2. Initial rotor position estimation (IRPE)

First, talking about the initial rotor position estimation, as the name states, these works are only focused on getting the initial position of the rotor by using different techniques. These techniques have not been discussed in detail, as this is not the objective of this work. However, in the reference [6], a description of an algorithm and how people have been trying to estimate initial position of the rotor only to be used as an initial condition for another estimator is provided.

Where rotor position estimation can be roughly divided into two categories [6]:

1. Fundamental excitation methods [7, 8]
    a. Flux linkage based
    b. Back electromotive force (EMF) based
    c. Sliding Mode Observer [23]
    d. Advanced techniques based
2. Saliency based methods
    a. Signal injection methods [9]
        i. Continuous signal injection
        ii. Transient voltage vector injection
        iii. High frequency signal injection
    b. Without signal injection methods
        i. PWM excitation [33]

These categories can have further subcategories as this is a very wide topic with a lot of possibilities to implement the estimation algorithms. Hence there are more estimation methods that may have different titles but at the end, they lie between one of these two main categories of the algorithms.

Summarizing, in some of the fundamental excitation methods, the rotor position is obtained by estimating the back electromotive force (EMF) based on the motor mathematical model. Observer methods [10-13] and the extended Kalman filtering method [14, 15] are used to extract the back EMF information. The parameter estimation scheme is combined with the position observer to guarantee the accuracy of the motor mathematical model, even if the

electrical parameters are influenced by the temperature factor [16]. These methods are effective in the medium to high speed region. However, at the low speed region, these methods fail to operate because the back EMF information is too small.

Rather than the back EMF, the signal injection methods are based on the anisotropic properties of PMSMs, caused either by the inherent saliency due to the machine geometry [17, 18] or by the saturation of the stator iron [19]. Therefore, these methods can be utilized to estimate speed and position information in the low speed region, and in some cases including zero speed. Two kinds of carrier signals; revolving high frequency carrier signal and pulsating high frequency carrier signal, are usually used to estimate the rotor speed and position [20-22]. On the other hand, the PWM excitation technique also has the drawback of not allowing the estimation in the whole range of the motor operation, they fail at the high speed. Distinguishing between 0-180 degrees and 180-360 degrees rotor position is a problem that requires new methods to solve it. As the observer needs to be implemented in the motor control loop it impacts the software throughput [33].

2.1. Three recent works as examples

2.1.1. Novel Sliding Mode Observer (NSMO)

Reference [23] presents a back EMF based Sliding Mode Observer (SMO), which is an advanced version of a conventional SMO. This work was published in 2018 3rd International Conference on Advanced Robotics and Mechatronics (ICARM).

Figure (2-1) represents the structural implementation of the control system.

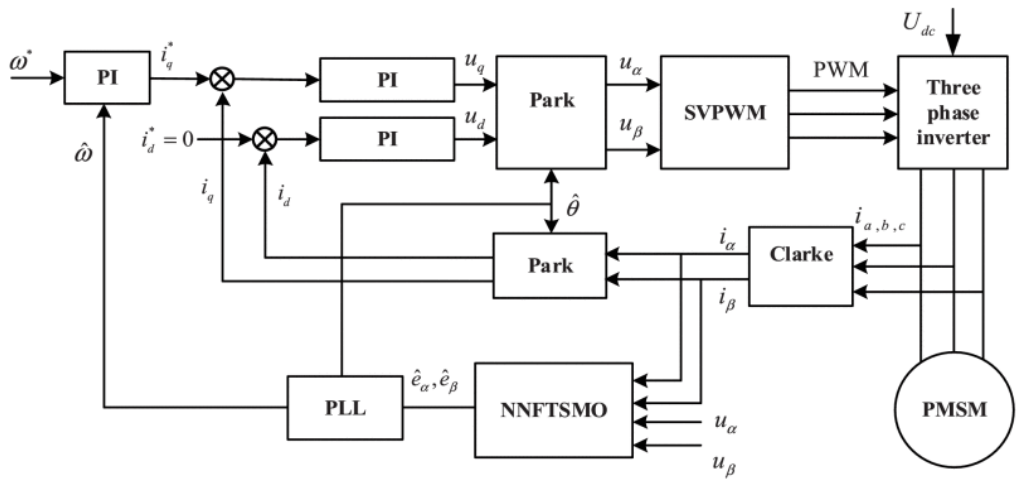


Figure 2-1: Novel sliding mode observer speed governing system

Based on the theory of SMC, a novel integral non-singular fast terminal sliding mode observer (NINFTSMO) is proposed in that work. The integral term can obviously reduce the noise and signal distortion from differential state and exponential factor can improve convergence speed. In addition, in order to restrict chattering, a novel sigmoid function is used instead of traditional sign function. Furthermore, the position information of the rotor is estimated from the back electromotive force by phase locked loop, which can improve observation accuracy. The stability of NINFTSMO is proved and demonstrated which is based on the Lyapunov stability theorem.

The two outputs of PLL are $\hat{\theta}$ is the estimated rotor position angle and $\hat{\omega}$ is the estimated speed. Figure (2-2) explores the observer only:

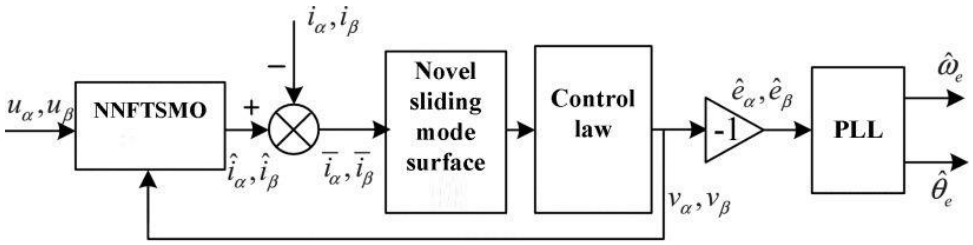


Figure 2-2: Proposed design of novel integral nonsingular sliding mode observer

Where the NNFTSMO, Novel sliding mode surface and the control law are implemented by the following equations (2.1), (2.2) and (2.3) respectively;

$$\left[\begin{array}{l}\frac{d\,\hat{i}_\alpha}{dx} = -\frac{R}{L}\,\hat{i}_\alpha + \frac{\hat{u}_\alpha}{L} + \frac{\widehat{\bar{v}_\alpha}}{L} \\ \frac{d\,\hat{i}_\beta}{dx} = -\frac{R}{L}\,\hat{i}_\beta + \frac{\hat{u}_\beta}{L} + \frac{\hat{\bar{v}}_\beta}{L}\end{array}\right] \quad (2.1)$$

$$s = x + m\int_0^t x^\delta d\tau + n\int_0^t x^\lambda \Gamma(x) d\tau \quad (2.2)$$

$$v = R_s i^- - L\left(m i^{-\delta} + n i^{-\lambda}\Gamma(i^-)\right) - \epsilon s - \eta \mathrm{sgn} s \quad (2.3)$$

The Phase Locked Loop (PLL) implementation is described in the Figure (2-3):

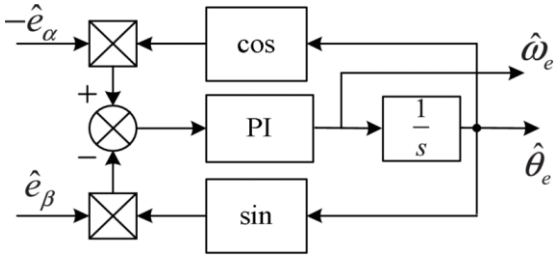


Figure 2-3: Phase locked loop structure

The authors of this work do not mention the maximum and minimum limits of speed on which this observer works. However, as we know already that this is a back-emf based observer and according to research that has been done regarding these topics, it is very obvious that this observer works for medium to high speeds only. Hence, it cannot fulfil the need of performing estimation on low and zero speed.

2.1.2. Initial position detection for PMSM

Reference [24] also presents a high frequency signal injection based method which is also a kind of observer. This work was published in 2018 IEEE 9th International Symposium on sensorless Control for Electrical Drives (SLED).

This method utilizes the machine anisotropy that is a function of the rotor position. The significant saliency can be observed in PMSM, which yields to different inductance value in d and q axes [26, 27]. This paper describes a saliency tracking method to estimate the rotor position. The magnetic saliency is excited with the injection of high frequency voltage signal. Figure (2-4) shows the saliency based observer designed in that work.

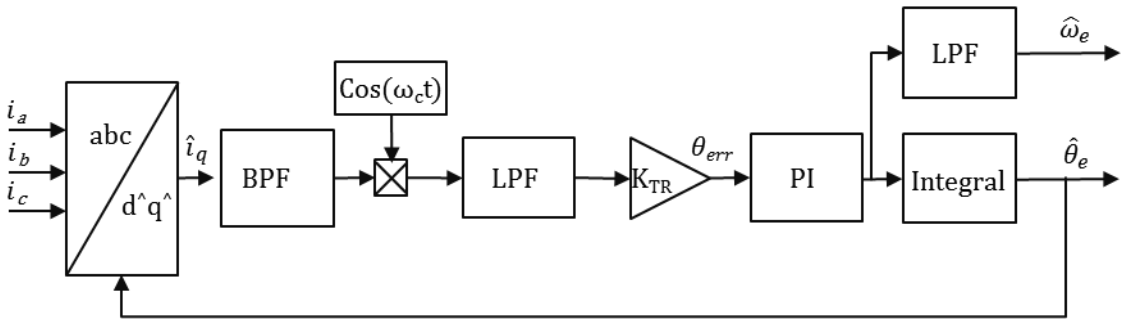


Figure 2-4: Proposed design of saliency observer

Where $\hat{\theta}_e$ is the estimated rotor position/angle and $\hat{\omega}_e$ is the estimated speed. The angle error, $\theta_{err,}$ is extracted by the following equations:

$$\begin{bmatrix} \hat{i}_d = -\frac{\hat{U}_m}{\omega_c L_d L_q}\cos(\omega_{ct})\,(L_o - \Delta_L \cos(2\theta_{err})) \\ \hat{i}_q = -\frac{\hat{U}_m}{\omega_c L_d L_q}\cos(\omega_{ct})\,(\Delta_L \sin(2\theta_{err})) \end{bmatrix} \quad (2.4)$$

ω_c is the carrier frequency of the injected voltage signal into the system. Therefore, this saliency based observer is a closed-loop system that forces the filtered component of the current $\hat{i}_q$ to be zero, and therefore setting θ_{err} to zero. In such case, the estimated dq frame matches the real dq frame and the estimated rotor position reflects the rotor flux position.

As stated before, this method uses injection of high frequency voltage signals in the system to determine the rotor position, thus reducing the overall efficiency of the system by increasing the losses in the machine, but the initial position detection has also been verified on a PMSM. Moreover, this algorithm can detect the initial rotor position [25, 28] and estimate the rotor position for low speeds and/or zero speed as stated by the authors.

2.1.3. High frequency injection rotor position estimation

Figure (2-5) describes the control system diagram for the presented work in [29].

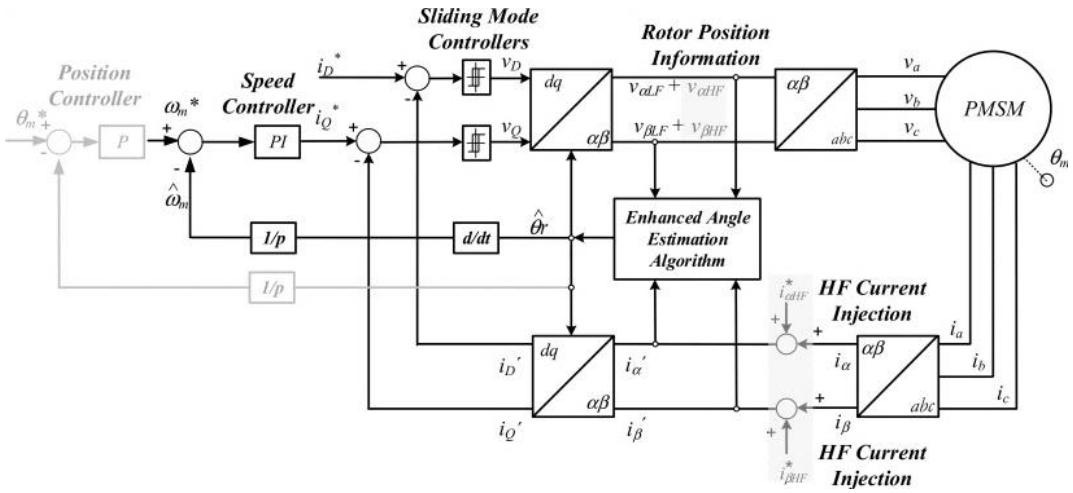


Figure 2-5: Enhanced angle estimation algorithm with control system block diagram

Reference [29] presents an enhanced method by using high frequency signal injection for PMSMs. This work was presented in 2014 IEEE 11th International Multi-Conference on Systems, Signals & Devices.

The novelty of this work is that the designed system works with the non-continuous control, with a SMC, unlike the conventional high frequency injection techniques, which only works with continuous control techniques.

The enhancement of the sensorless algorithm is achieved when the non-desired component described in equation (2.6) is minimised. Notice that the perturbing voltages of equation (2.6) are produced by the propagation of the estimated angle error through the machine from the signals i'_α and i'_β, dependent on the estimated angle as equation (2.5) indicates, to the voltages v_α and v_β.

$$\begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix} = \begin{bmatrix} i'_\alpha(\hat{\theta}_r) - i^*_{\alpha HF} \\ i'_\beta(\hat{\theta}_r) - i^*_{\beta HF} \end{bmatrix} = \begin{bmatrix} -i^*_Q \sin(\hat{\theta}_r) - A_p \sin(\omega_i t) \\ i^*_Q \cos(\hat{\theta}_r) + A_p \cos(\omega_i t) \end{bmatrix} \quad \textbf{(2.5)}$$

$$e_{v\alpha} = [i^*_Q \frac{d\hat{\theta}_r}{dt}(L_s \cos(\hat{\theta}_r) - \Delta L_s \cos(\theta_r - e_{es}))]_{BP}$$
$$e_{v\beta} = [i^*_Q \frac{d\hat{\theta}_r}{dt}(L_s \sin(\hat{\theta}_r) - \Delta L_s \sin(\theta_r - e_{es}))]_{BP} \quad \textbf{(2.6)}$$

The proposed enhanced algorithm is described in following block diagram; Figure (2-6).

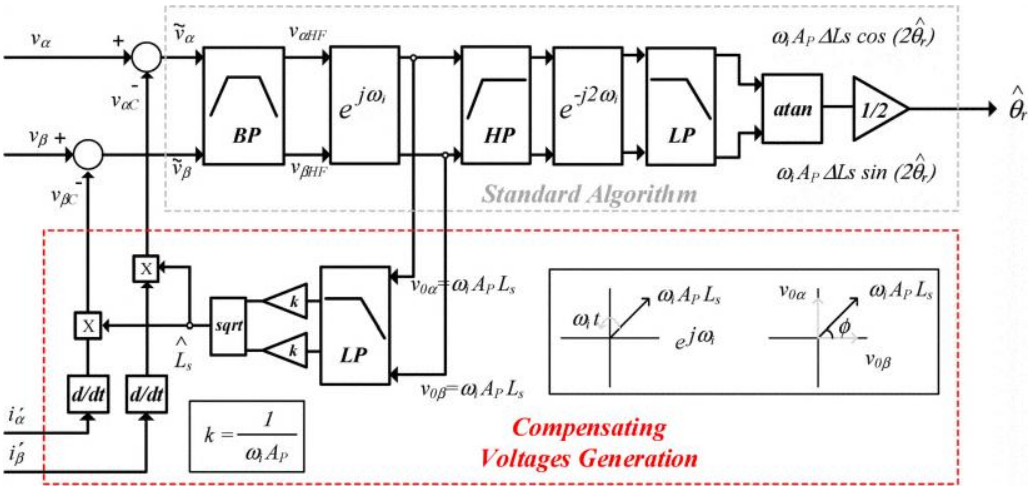


Figure 2-6: Proposed design of enhanced angle estimation technique

This algorithm is not being tested in the real system yet. However theoretically and with the help of simulations the implementation of it has been verified. Nevertheless, it is able to estimate the rotor angle position for PMSMs for low and zero speeds, based on the high frequency injection technique to track the machine saliency. The novelty of the work relies on the fact that the PMSM drive is under SMC. However, the injection of high frequency signals into the machine causes increment of losses. Hence, it results in low efficiency of the overall system.

3. Design of zero speed rotor position estimator based on SMC for PMSM

The idea of this estimator comes with the implementation of a decoupled SMC control (which is designed by neglecting the saliency effect) on a PMSM and then analysing the obtained sliding surfaces signals considering the actual saliency effect which proves the theoretical possibility to design such estimator.

Hence a SMC has been designed and implemented on PMSM. Furthermore this control has been designed in abc framework with well-known decoupling method and it has been experimented in a real system with nominal conditions [30].

3.1. Decoupled SMC applied to PMSM

According to the theoretical design, once we have decoupled sliding surfaces under sliding motion, if there is no saliency in the motor, the decoupled sliding surfaces must behave as piecewise linear functions as shown in the switching surfaces graph presented in Figure (3-1):

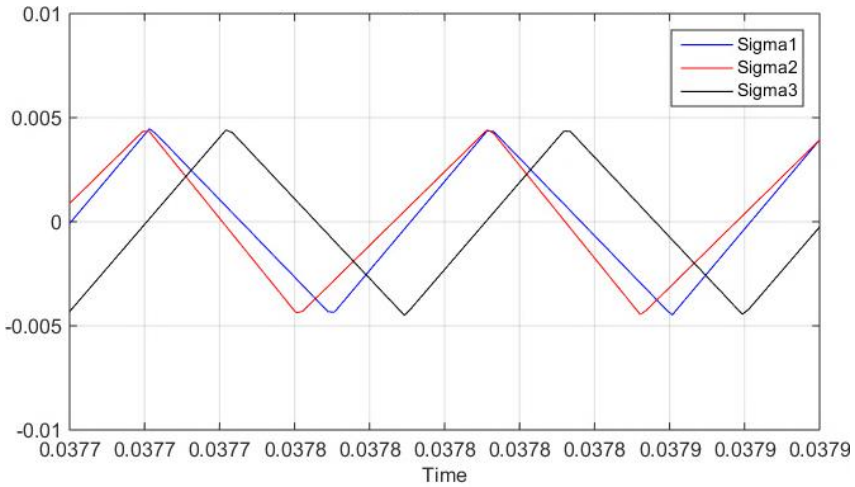


Figure 3-1: Switching surfaces without saliency effect

Furthermore, in the design criteria, the motor saliency effect is assumed as zero because, normally, saliency has much lower value as compared to the inductances of the motor, thus it can be neglected. On the other hand, by using decoupled controls, it is possible to implement variable hysteresis band comparators to get fixed switching frequency operation [30].

The SMC design steps neglecting saliency are summarized in the following steps:

First, recalling that in a balanced 3 phase system, the algebraic equation, $i_a+i_b+i_c = 0$ holds due to the start connection of the stator windings, which is simplified to:

$$L'_m \frac{di}{dt} = v_{abcn} - \left(RI + \frac{dL}{dt}\right) i - e(\omega_e, \theta_e) \quad \textbf{(3.1)}$$

where; $L'_m = (3/2)\, Lm.$

On the other hand, the stator voltages are given by:

$$v_{abcn} = [v_a - v_n, v_b - v_n, v_c - v_n]^T \quad \textbf{(3.2)}$$

Being $v = [v_a, v_b, v_c]^T$ the voltages produced by the voltage source inverter (VSI) and v_n is the voltage at the neutral point of the machine as shown in Figure (3-4). These voltages are related to the VSI control signals $u = [u_a, u_b, u_c]^T$ as follows:

$$v = v_{bus}\, u \qquad (3.3)$$

where v_{bus} is the DC bus voltage applied to the VSI to drive the motor. The VSI control actions u_a, u_b and u_c can only take the discrete values {-1, 1}.

By merging equations (1.9), (3.1) and (3.3), we get the motor dynamics that includes the control signals of the VSI and it can be rearranged as follows:

$$L'_m \frac{di}{dt} = -\left(RI + \frac{dL}{dt}\right) i - e(\omega_e, \theta_e) + v_{bus}\, Bu \qquad (3.4)$$

where;

$$B = \frac{1}{3}\begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \qquad (3.5)$$

The main objective of the control is to achieve the proper tracking of the current references. Due to the balanced system and algebraic constraint given by $i_a+i_b+i_c = 0$ (that implies that B is not invertible), the switching surfaces, that are proposed in the next given equation (3.6), control the currents i_a, i_b and the third control signal is used to control and equilibrate the system by regulating the average of v_n to 0. Hence the proposed switching surfaces for the motor are:

$$S = [s_a, s_b, s_c]^T = [i_a^* - i_a, i_b^* - i_b, \int (v_n^* - v_n) dt]^T \qquad (3.6)$$

where the superscript "*" stands for the reference values. The switching functions time derivative becomes as follows:

$$S' = f - v_{bus}\, C\, u \qquad (3.7)$$

where;

$$C = \frac{1}{3L'_m}\begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ L'_m & L'_m & L'_m \end{bmatrix}; \quad f = \begin{bmatrix} i_a^{*\prime} + \frac{R}{L'_m} i_a + \frac{e_a}{L'_m} \\ i_b^{*\prime} + \frac{R}{L'_m} i_b + \frac{e_b}{L'_m} \\ v_n^* \end{bmatrix} \qquad (3.8)$$

According to the equation (3.6), to design a decoupled control, a new set of surfaces through a decoupling process must be designed. Therefore, the new set of surfaces can be described as follows:

$$\sigma = MS \qquad (3.9)$$

where $\sigma = [\sigma_a, \sigma_b, \sigma_c]^T$ is the new set of surfaces and M is a non-singular matrix that is obtained by solving the equations (3.1), (3.7) and (3.9). Therefore, the decoupling matrix has the following elements:

$$M = C^{-1} = \begin{bmatrix} L'_m & 0 & 1 \\ 0 & L'_m & 1 \\ -L'_m & -L'_m & 1 \end{bmatrix} \qquad (3.10)$$

Hence, according to the definition of the matrix M, the first time derivative of σ becomes;

$$\dot{\sigma} = \mathrm{Mf} - \mathrm{v_{bus}}\,\mathrm{MCu} = \mathrm{Mf} - \mathrm{v_{bus}}\,\mathrm{Iu} \quad \textbf{(3.11)}$$

where;

$$\mathrm{Mf} = \begin{bmatrix} \mathrm{L'_m i_a^{*'}} + \mathrm{Ri_a} + \mathrm{e_a} + \mathrm{v_n^*} \\ \mathrm{L'_m i_b^{*'}} + \mathrm{Ri_b} + \mathrm{e_b} + \mathrm{v_n^*} \\ -\mathrm{L'_m}(\mathrm{i_a^{*'}} + \mathrm{i_b^{*'}}) + \mathrm{Ri_c} + \mathrm{e_c} + \mathrm{v_n^*} \end{bmatrix} \quad \textbf{(3.12)}$$

The finite time convergence of both σ and $\mathbf{S}$ vectors to 0 is possible by means of a Lyapunov function candidate $\mathrm{V} = 0.5\mathrm{S^T S}$, and designing the control such that $\dot{\mathrm{V}} < 0$ [30], where control signals are selected according to the following law:

$$\mathrm{u_i} = \mathrm{sign}(\sigma_\mathrm{i}),\ \mathrm{i} \in \{\mathrm{abc}\} \quad \textbf{(3.13)}$$

3.2. Analysis of the saliency effects in SMC

3.2.1. PMSM saliency

Saliency is the state or quality by which something stands out relative to its neighbour elements. As it is described before that the presented estimator is based on the saliency effect of the motor or to be precise, it extracts the rotor position information from the effect of saliency. Magnetic saliency describes the relationship between the rotor's main flux (d axis) inductance and the main torque producing (q axis) inductance. The magnetic saliency varies depending on the position of the rotor to the stator field, where maximum saliency occurs at 90 electrical degrees from the main flux axis (d axis). Depending on the motor structure, the value of the saliency could vary, but as mentioned before, saliency value is very small if compared with the inductances of the motor, thus it is often neglected when designing a control.

Figure (3-2) shows d and q axes almost 90 degrees apart in dq frame:

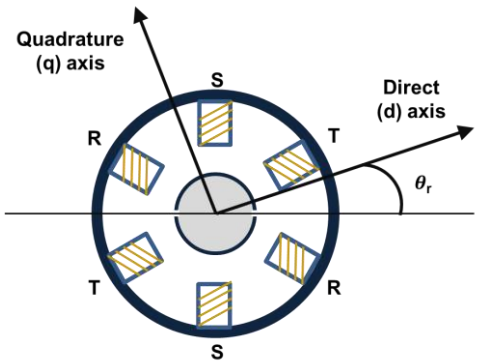


Figure 3-2: Direct Quadrature (dq) axes representation along the rotor of a PMSM

In which, d and q axes are moves along with rotor with an angle θ_r.

3.2.2. Effects of the saliency in the system under SMC

However in a real system, the saliency parameter cannot be controlled and assumed as zero, it is always there with a specific value when applying a SMC, whether or not the control is designed with the compensation for the saliency effect.

Hence when applied to a real system or in a system where saliency effect is not zero (simulation), the changes in the signal slopes can be observed as indicated by the red arrows in the Figure (3-3):

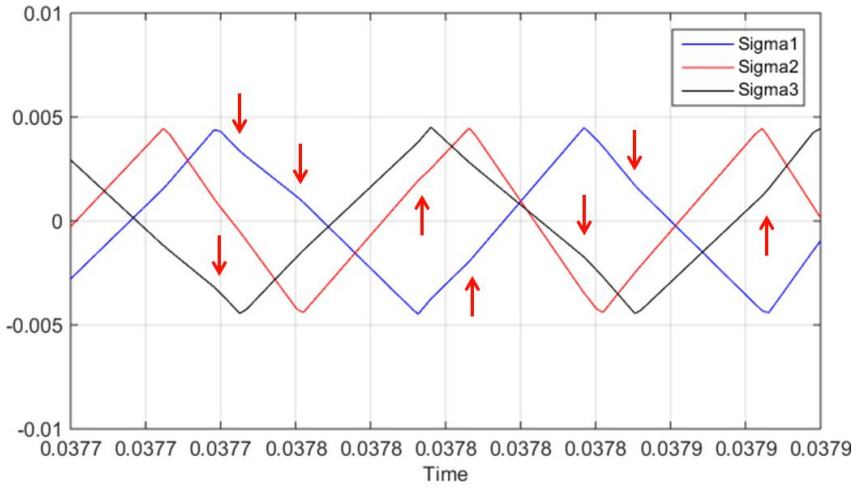


Figure 3-3: Switching surfaces with saliency effect

If compared with Figure (3-1), it can be easily verified that the sliding surfaces are no more piecewise linear, and this is due to saliency effect, thus, these changes in slopes have made it possible to extract the information about the rotor position and to design this rotor position estimator for low and zero speeds. As we know a real motor will always have saliency, whether the motor is moving or not, thus there is no need to inject additional signals, as long as control is there, the rotor position is estimated all the time. This algorithm needs initial condition of the rotor at the beginning and once the system in under sliding motion, this estimator works perfectly within the desired and designed range of speed.

Remark III: *This algorithm still requires initial position as initial condition of the rotor in the beginning, which in this case, has been taken from the encoder attached to the motor. In literature, there has been done a lot of research only on the topic of getting the initial condition for the estimation of rotor. Researchers are still trying new methods just to estimate the initial position of the rotor [24, 28].*

On the other hand, some features of this algorithm have been analysed to get the initial condition of the rotor. It has been observed that it is possible to get the initial condition of the rotor by using some parameters under sliding motion, but it needs to be verified in the future.

Proceeding to redesign the calculations made in “Decoupled SMC applied to PMSM”, the small term of saliency that was neglected before will produce the effects such that the matrix M does not perfectly decouple the switching functions. This fact will produce the changes in the slopes of the switching functions depending on the vector control state, as described before as nonlinear behaviour of the switching functions.

Remark IV: *This does not produce any impact from the control point of view on the system, as it will be shown later, as long as the sliding motion is guaranteed under these conditions.*

Nevertheless, such slopes deviations can be exploited to extract the information from the system, in this case to estimate the rotor angle.

Assuming ΔL is not zero and starting with the PMSM model equation (1.1):

$$L\frac{di}{dt} = -\left(RI + \frac{dL}{dt}\right)i - e(\omega_e, \theta_e) + v_{bus}\, B\, u \qquad (3.14)$$

Due to the restriction $i_a+i_b+i_c=0$, the system becomes:

$$\begin{bmatrix} \frac{di_a}{dt} \\ \frac{di_b}{dt} \end{bmatrix} = -L_z^{-1}\left(\left(RI + \frac{dL_z}{dt}\right)\begin{bmatrix} i_a \\ i_b \end{bmatrix} + \begin{bmatrix} e_a \\ e_b \end{bmatrix}\right) + v_{bus}\, B_z\, u \qquad (3.15)$$

where

$$L_z = \begin{bmatrix} L_a - L_{ac} & L_{ab} - L_{ac} \\ L_{ba} - L_{bc} & L_b - L_{bc} \end{bmatrix} \qquad (3.16)$$

$$L_z^{-1} = \frac{4}{9(L_m^2 - \Delta_L^2)}\begin{bmatrix} L_b - L_{bc} & L_{ac} - L_{ab} \\ L_{bc} - L_{ba} & L_a - L_{ac} \end{bmatrix} \qquad (3.17)$$

$$B_z = L_z^{-1}\left(\frac{1}{3}\right)\begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \end{bmatrix} \qquad (3.18)$$

According to equation (3.6), the first time derivative of the switching surface results in:

$$\begin{bmatrix} \dot{S}_a \\ \dot{S}_b \\ \dot{S}_c \end{bmatrix} = \begin{bmatrix} \frac{di_a^*}{dt} \\ \frac{di_b^*}{dt} \\ \frac{dv_n^*}{dt} \end{bmatrix} + L_s\begin{bmatrix} i_a \\ i_b \\ 0 \end{bmatrix} + \begin{bmatrix} L_z^{-1} \\ 0 \end{bmatrix}\begin{bmatrix} e_a \\ e_b \\ 0 \end{bmatrix} - v_{bus}\, C_s\, u \qquad (3.19)$$

where L_s and C_s are defined as:

$$C_s = \frac{1}{3}\begin{bmatrix} 0 & 3B_z & 0 \\ 1 & 1 & 1 \end{bmatrix},\ L_s = \begin{bmatrix} L_z^{-1}(RI + \frac{dL_z}{dt}) & 0 \\ 0 & 0 \end{bmatrix} \qquad (3.20)$$

Therefore (3.19) can be rewritten as:

$$\dot{S} = f_1 - v_{bus}\, C_s\, u \qquad (3.21)$$

where f_1 contain all the terms that do not depend on the control. Finally, the time derivative of the decoupled surfaces becomes the following:

$$\dot{\sigma} = Mf_1 - v_{bus}\, MC_s\, u = f_2 - v_{bus}\, MC_s\, u \qquad (3.22)$$

In which, the term f_2 is related to the low frequency components of the system, and hence, to the equivalent control [31]. These terms are dependent on the motor parameters, and the working conditions and it can be bounded for certain conditions. From which the term depending on the control is $v_{bus}\ MC_s$ being v_{bus} a known value. The product of matrices MC_s produces a full matrix (where in the case of no saliency it is a diagonal matrix), and the switching functions slopes are influenced by all the control signals. The result for $\dot{\sigma}$ can be written as follows:

$$\dot{\sigma}_i = f_{2i} - v_{bus}\, u_i + v_{bus}(A_1\, \delta_i + A_2\, \delta_{2i});\quad i \in \{abc\} \qquad (3.23)$$

where

$$A_1 = \frac{2\Delta_L L_m}{3(L_m^2 - \Delta_L^2)}\ ;\ A_2 = \frac{\Delta_L^2}{3(L_m^2 - \Delta_L^2)} \qquad (3.24)$$

and the values of δ_i and δ_{2i} are shown in the Table 2:

Table 2: Expected terms for δ and δ2

	σ_a
δ_a	$\cos(2\theta_e)u_a - \cos\left(2\theta_e - \frac{2\pi}{3}\right)u_b - \cos\left(2\theta_e + \frac{2\pi}{3}\right)u_c$
δ_{2a}	$[u_b + u_c - 2u_a]$
	σ_b
δ_b	$\cos(2\theta_e)u_c - \cos\left(2\theta_e + \frac{2\pi}{3}\right)u_b - \cos\left(2\theta_e - \frac{2\pi}{3}\right)u_a$
δ_{2b}	$[u_a + u_c - 2u_b]$
	σ_c
δ_c	$\cos(2\theta_e)u_b - \cos\left(2\theta_e - \frac{2\pi}{3}\right)u_c - \cos\left(2\theta_e + \frac{2\pi}{3}\right)u_a$
δ_{2c}	$[u_a + u_b - 2u_c]$

Notice that, the switching functions slopes contains the terms that are dependent on Δ_L and $2\theta_e$.

Now, let us check the conditions that guarantee the convergence of σ to 0. To prove that the following Lyapunov function candidate has been selected:

$$V = 0.5\sigma^T \sigma \qquad \textbf{(3.25)}$$

with time derivative;

$$\dot{V} = \sigma^T f_2 - v_{bus}\sigma^T M C_s\, u \qquad \textbf{(3.26)}$$

This is equivalent to;

$$\dot{V} = \sum_{i=a,b,c}\sigma_i f_{2i} + v_{bus}\sum_{i=a,b,c}\sigma_i(A_1\,\delta_i + A_2\,\delta_{2i}) - v_{bus}\sum_{i=a,b,c}\sigma_i u_i \qquad \textbf{(3.27)}$$

From Table 2, the following bounds can easily be found:

$$\delta_i < \delta_M = 2, \qquad \delta_{2i} < \delta_{2M} = 4, \qquad \forall i \in \{abc\}$$

Furthermore, by using the control law defined before, u_i=sign (σ_i), i∈{abc}, and taking into account that the values f_{2a}, f_{2b} ,f_{2c} can be bounded by f_{2aM}, f_{2bM} ,f_{2cM}, one can get:

$$\dot{V} \le \sum_{i=a,b,c}|\sigma_i| f_{2i}M + \epsilon v_{bus}\sum_{i=a,b,c}|\sigma_i| \qquad \textbf{(3.28)}$$

Where $\epsilon = 1 - \delta_M A_1 - \delta_{2M} A_2$.

Therefore, selecting a bus voltage that fulfils the following inequality:

$$v_{bus} > \frac{\max(f_{2aM}, f_{2bM}, f_{2cM})}{\epsilon} + \eta,\ \eta > 0 \qquad \textbf{(3.29)}$$

It can be ensured that the derivative of the Lyapunov function is negative and the desired convergence is achieved in finite time.

Remark V: *According to the data shown in Table 4, the term $\delta_M A_1 - \delta_{2M} A_2$ becomes 0.27 for this example, which implies a bus voltage that is 36% higher than the one in the non-saliency case.*

3.3. Zero speed rotor position estimation algorithm

The angle estimation technique can be derived from the result obtained in the following equation:

$$\dot{\sigma}_i = f_{2i} - v_{bus}\, u_i + v_{bus}(A_1\, \delta_i + A_2\, \delta_{2i}) \quad \textbf{(3.30)}$$

The proposed methodology is based on the equivalent control technique [31]. Hence by applying the equivalent control method we get:

$$0 = f_{2ieq} - v_{bus}\, u_{ieq} + v_{bus}(A_1\, \delta_{ieq} + A_2\, \delta_{2ieq}) \quad \textbf{(3.31)}$$

where δ_{ieq} and δ_{2ieq} arise replacing the values of u_a, u_b and u_c by their equivalent control values in δ_i and δ_{2i}, respectively. Hence;

$$f_{2ieq} = v_{bus}\, u_{ieq} - v_{bus}(A_1\, \delta_{ieq} + A_2\, \delta_{2ieq}) \quad \textbf{(3.32)}$$

On the other hand, the control law presented in equation (3.13), implies infinite switching frequency, which cannot be attained in real implementations. As from reference [30], those control laws were replaced by comparators with hysteresis, obtaining finite switching frequency of the control actions. In this case, while in the real sliding motion, it holds the condition $|\sigma_i| < \Delta$, being Δ the hysteresis width.

Now, if the switching frequency is high enough and the chosen value of hysteresis band produces small oscillations of σ_i in the vicinity of surfaces $\sigma_i = 0$, the vector f_{2i} can be approximated by f_{2ieq}. Thus;

$$f_{2i} \approx f_{2ieq} \quad \textbf{(3.33)}$$

Therefore, by putting equation (3.32) in (3.23) we get:

$$\dot{\sigma}_i = v_{bus}(u_{ieq} - u_i) + v_{bus}[A_1(\delta_i - \delta_{ieq}) + A_2(\delta_{2i} - \delta_{2ieq})] \quad \textbf{(3.34)}$$

The above expression is the basis for the estimation.

Measuring σ'_i, u_{ieq}, knowing the v_{bus} and the vector control state, it is possible to calculate:

$$\alpha_i = \frac{\dot{\sigma}_i - v_{bus}(u_{ieq} - u_i)}{v_{bus}} = A_1(\delta_i - \delta_{ieq}) + A_2(\delta_{2i} - \delta_{2ieq}) \quad \textbf{(3.35)}$$

The estimation is performed assuming that the terms δ_i and δ_{2i} are much bigger than δ_{ieq} and δ_{2ieq}. This fact has been proven with numerical simulations and later it has been tested in a real system. More details about it can be found in the further Chapters. Hence;

$$\alpha_i \approx A_1\delta_i + A_2\delta_{2i} \quad \textbf{(3.36)}$$

The terms δ_i and δ_{2i} depends on control vectors as described in the Table 2. The discrete values {-1, 1} of each control signal generate eight different control combinations for δ_i and δ_{2i} as shown in the Table 3.

In which;

$$x = 2\theta_e, \qquad y = 2\theta_e - \frac{2\pi}{3}, \qquad z = 2\theta_e + \frac{2\pi}{3}$$

It can be seen from the Table 3 that there are only six combinations where θ_e can be estimated. Notice that for any of these combinations, the signals $\cos(x)$, $\cos(y)$ and $\cos(z)$

are available. With these signals the estimation of θ_e is completely determined by using a control selector and some trigonometric properties.

Table 3: Control vectors and expected terms for δi and δ2i

u_a	u_b	u_c	δ_a	δ_{2a}	δ_b	δ_{2b}	δ_c	δ_{2c}
-1	-1	-1	0	0	0	0	0	0
-1	-1	1	2cos(z)	-2	2cos(x)	-2	2cos(y)	4
-1	1	-1	2cos(y)	-2	2cos(z)	4	2cos(x)	-2
-1	1	1	−2cos(x)	-4	−2cos(y)	2	−2cos(z)	2
1	-1	-1	2cos(x)	4	2cos(y)	-2	2cos(z)	-2
1	-1	1	−2cos(y)	2	−2cos(z)	-4	−2cos(x)	2
1	1	-1	−2cos(z)	2	−2cos(x)	2	−2cos(y)	-4
1	1	1	0	0	0	0	0	0

A general block diagram of the implemented estimator with the control system is shown in the Figure (3-4);

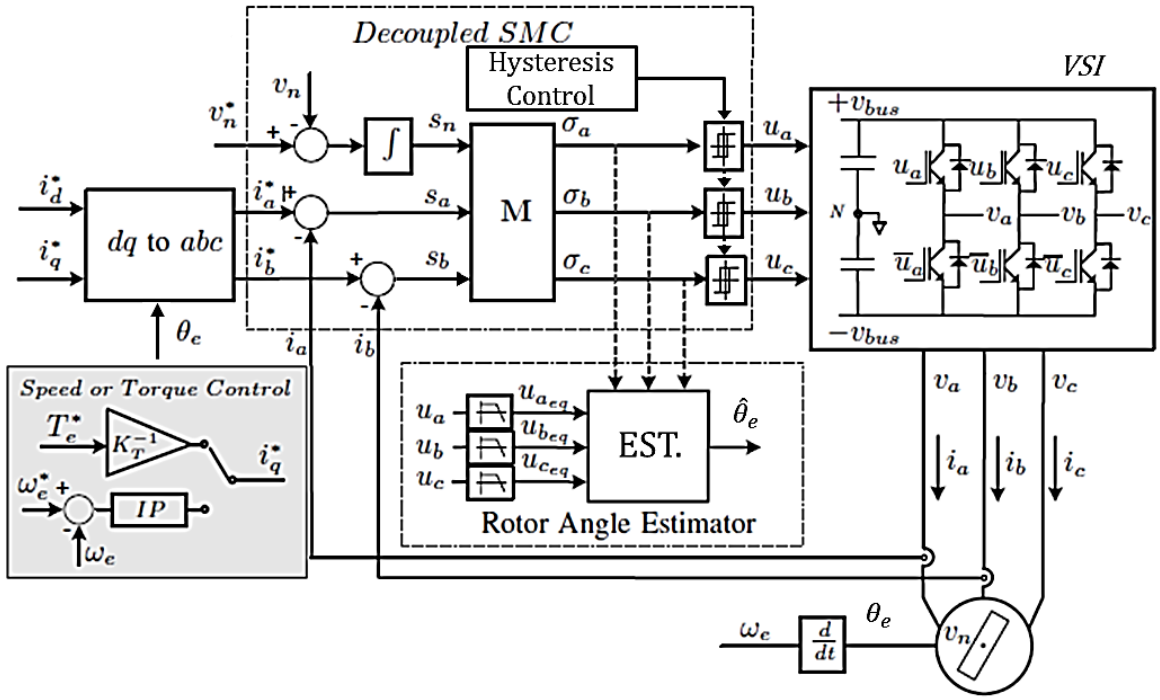


Figure 3-4: Decoupled SMC full scheme with rotor angle estimator

The system can be controlled by either with speed control or with torque control. In this work, mostly speed control is used to perform all the simulations.

Therefore, the proposed rotor angle estimator is implemented as shown in the Figure (3-5), on the next page.

In which, to measure the change in slopes, $\dot{\sigma}_i$, we use:

$$\dot{\sigma}_i \cong \frac{\sigma_{ik} - \sigma_{i(k-1)}}{T_s} \qquad (3.37)$$

Where the subscripts k, $k\text{-}1$ defines the sample values of σ and Ts is the sampling period.

The algorithm assumes a piecewise linear behavior of the switching functions for a constant value of the control vector.

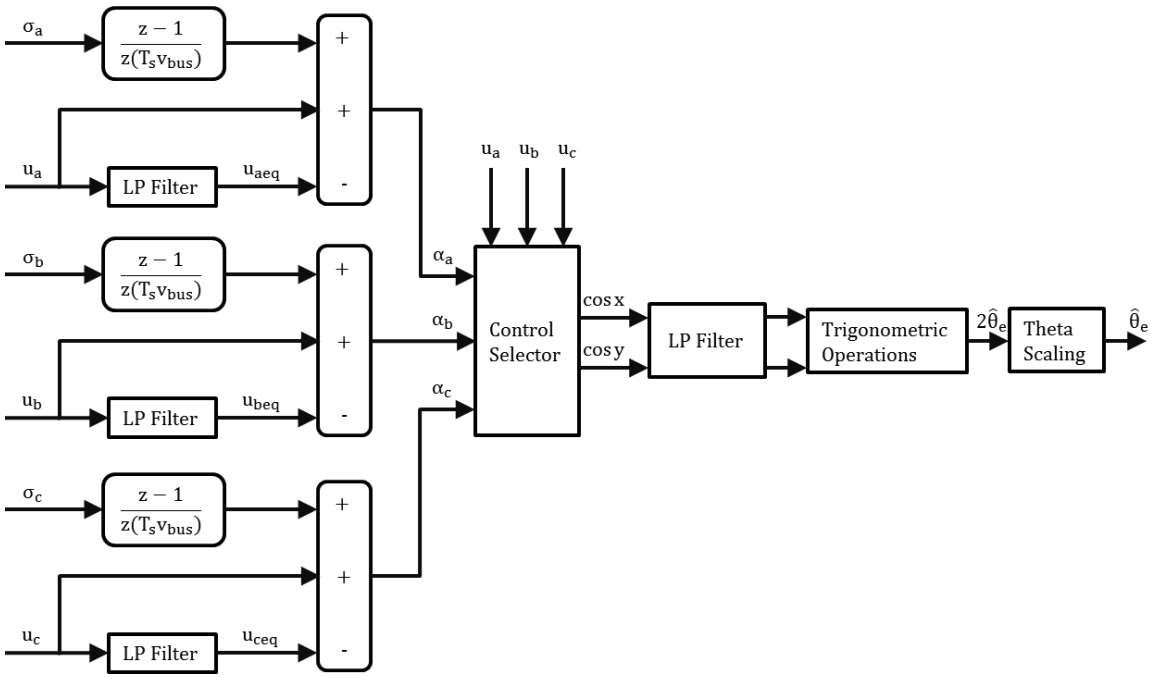


Figure 3-5: Proposed rotor angle estimator algorithm block diagram

The control selector is implemented as depicted in Table 3 to obtain the position signals, and trigonometric operations block is implemented as described in the Figure (3-6):

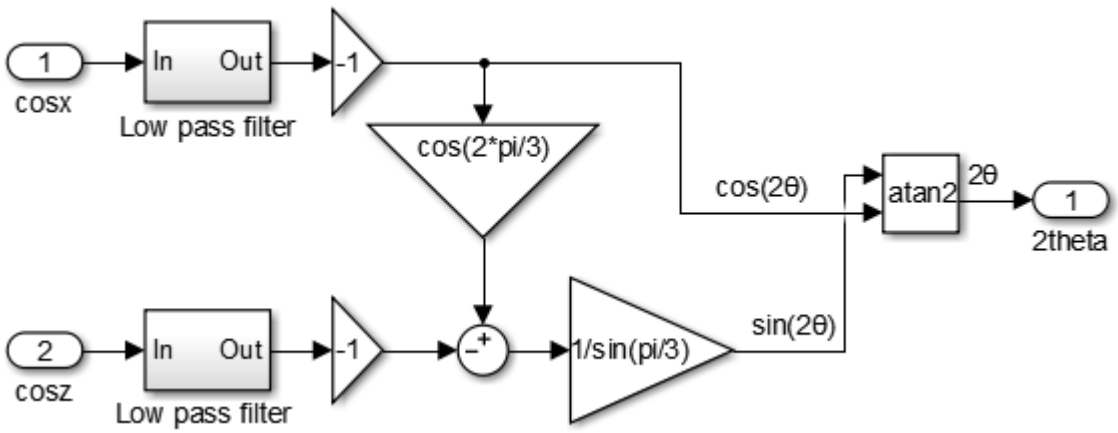


Figure 3-6: Implementation of trigonometric block with LP filter at the inputs

It can be observed that this block is generating cosine signal and sine signal (by performing some additional operations) to perform an arctangent to generate the $2\theta_e$ information from the position signals.

And lastly the theta scaling block is converting the $2\theta_e$ information to θ_e by performing some additional mathematical operations which are implemented as shown in the Figure (3-7) on the next page. Input signal to this block is $2\theta_e$ signal mapped between $-\pi$ and π, as obtained from the arctangent function from trigonometric operations block. As we know that conventionally the angle of the machine is always mapped between 0 to 2π. Thus, the first operation in the scaling maps the input $2\theta_e$ signal between 0 and π.

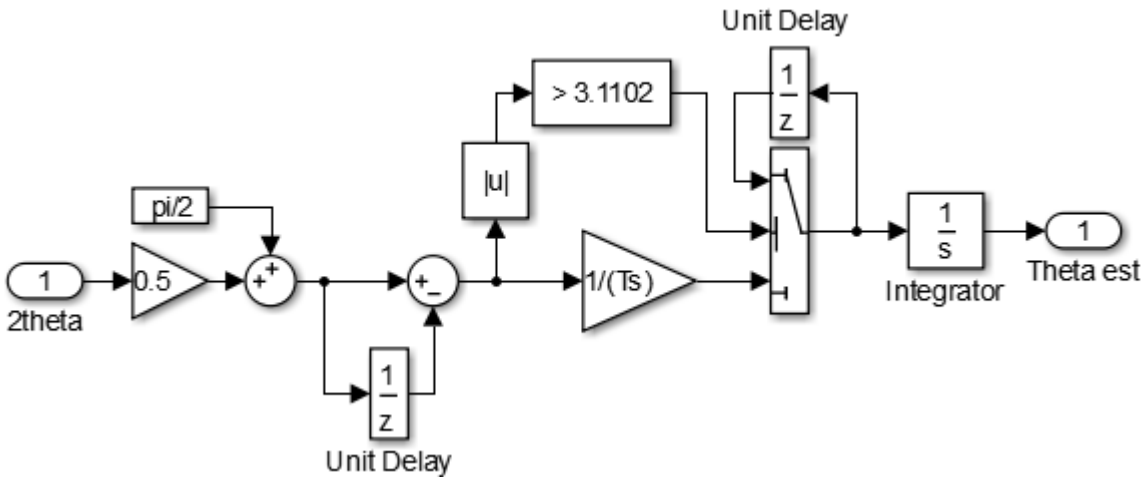


Figure 3-7: Theta scaling operations from $2\theta_e$ to θ_e block diagram

Then the $2\theta_e$ information is converted into speed information by using the relation $\omega_e = d\theta_e/dt$. Then a switch circuit is implemented such that it only actives when the input signal; which is the difference of the two consecutive samples, is greater than the specified value, to avoid the points where the derivative tries to reach infinity, so with the help of this switch the unwanted spikes in the output signal are removed. Therefore, when the switch is activated, the output signal follows the last value and keeps it. Later notice that an integrator is used to generate $\hat{\theta}e$ from the estimated speed signal. So, the output of this integrator is either an ascending or a descending ramp or a constant value (in case if a motor is not moving). Hence the output of the integrator is mapped between 0 to 2π which provides the theta estimation.

3.4. Switching frequency analysis of the proposed control

The control law, u_i=sign (σ_i) where i∈{abc}, as designed in equation (3.13), theoretically assumes infinite switching frequency of the control actions. Under this assumption, the system trajectories are confined to the manifold $\sigma = 0$. From the implementation point of view, the switching frequency must be bounded. The most common application of the SMC in practical system is accomplished by means of hysteresis comparators instead of sign function [30]. In this case the control laws are rewritten as follows:

$$u_i = \begin{cases} -1, & \text{if } \sigma_i < -\Delta_i \\ 1, & \text{if } \sigma_i > \Delta_i \end{cases} \quad i \in \{a, b, c\} \qquad \textbf{(3.38)}$$

and all the trajectories and not confined to the manifold $\sigma = 0$, but oscillate around it within a boundary layer $2\Delta > 0$. Therefore, this boundary can be fixed hysteresis value or a variable hysteresis value that is designed according to [30].

3.4.1. Hysteresis control

To control the comparators to generate the control actions for the VSI, as mentioned before, two types of hysteresis controls have been implemented in the system.

1. Fixed hysteresis control
2. Variable hysteresis control

In fixed hysteresis control a fixed value is used for the hysteresis band in which the switching period can vary with motor parameters, stator currents, back-emf voltage and

speed [30]. On the other hand, in variable hysteresis control the hysteresis band is adjusted to control the switching period of the signals according to the design in [30].

The estimator has been simulated, tested and verified with both fixed and variable hysteresis controls. The simulation results obtained can be seen in the Chapter 4 and Chapter 5 and the experimental results can be seen in the Chapter 7.

4. Algorithm implementation in continuous time

To validate the estimator, the first simulations are performed in continuous time by using an abc model of the PMSM and assuming that there are no disturbances and system works in an ideal state.

4.1. Case study

To develop the simulations in order to validate the implementation of this estimator, a case study has been defined. The Figure (4-1) shows the reference speed for the simulation case study:

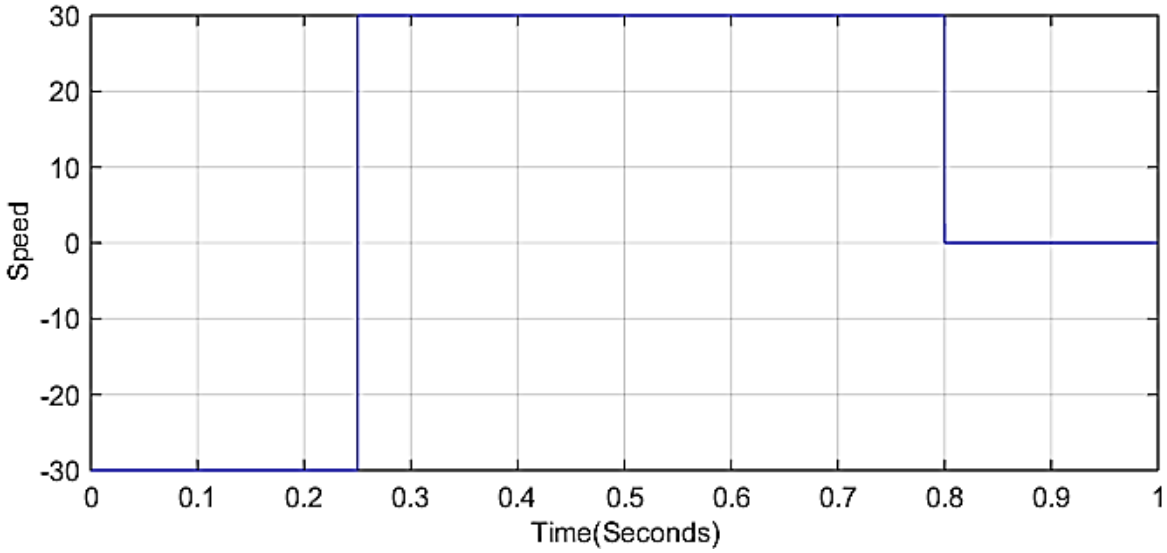


Figure 4-1: Reference speed plot for proposed case study

The system starts with a negative maximum speed (designed for this estimator only), -30 rad/s, the worst-case scenario intended for this estimator and then at time 0.25 s there is a speed reversal from -30 rad/s to 30 rad/s, then at 0.5 s there is load impact and at time 0.8 s speed is set to zero.

On the other hand, the graphs presented for this case study shows the following results:

1. Electrical speed: reference and real only, not estimated (in rpm)
2. Motor currents for all three phases (in amperes)
3. Theta: real and estimated (in rad/s)
4. Theta error (in degrees)

The simulations that are performed in this Chapter 4 and in the next Chapter 5, the first simulation in both chapters depict the complete case study result. Later zoomed graphs across all the transients for both open loop and sensorless operations are presented. Also, these simulations are made in the same graphical layout for the convenience of the reader to locate the graphs easily with possible comments.

Position signals estimation and speed estimation is performed in separate graphs for the same case study and the results have been presented in these Chapters.

Note I: *Speed estimation is not an objective of this thesis work. Nevertheless, it is performed only to compare the simulation results with the real system results without any*

additional improvements. Therefore, the obtained results from both cases: simulations and real system on oscilloscope are kept and compared in the further Chapters.

4.2. PMSM drive parameters

Table 4 describes the parameters of the motor drive that have been used to perform the simulations;

Table 4: Motor drive parameters

Parameter	Symbol	Value
Power	P	2.54 kW
Torque	T_e	8.1 Nm
Speed	ω_e	3000 rpm
Motor constants	K_T,K_e	0.93 Nm/A, 57 V/krpm
Pole pairs	P	3
Stator inductances	L_a, L_b, L_c	1.5 mH ± 0.25 mH
Stator resistance	R	0.36 ohm
Moment of inertia	J	4.57 10^{-3} kgm^2
Friction coefficient	B	8.57 10^{-3} Nms
Bus voltage	v_{bus}	135 V

Additionally, these are the same parameters as the physical motor drive that are used to perform the tests on the real hardware.

Important Note: *As a matter of fact, the simulation parameters for speed controller, hysteresis value and decimation have been chosen slightly different for continuous and discrete time simulations. E.g. in discrete time the decimation value has been chosen 10000, which is quite big, where in continuous time it is only 17, but it is to avoid the simulation crash during the discrete time simulation while saving a lot of information and performing the same case study in 1 s. Therefore, some information is lost but still reasonable results have been achieved for the comparison. Thus, in the following two Chapters, the simulation parameters have been clarified and due to that difference, the results presented in those Chapters, in terms of current ripples and speed transients are different. However, in average, talking about currents, we are looking for the same value because the motor parameters and applied load value are the same in both cases.*

4.3. Simulation data

Following Table 5 on next page shows the parameters used for these simulations:

Table 5: Simulation parameters for continuous time

Parameter	Value
Simulation step time, T_s	100 ns
Simulation time	1 s
Speed controller proportional gain, k_p	0.15
Speed controller integral gain, k_i	5
Rotor initial position	π = 3.1415
Bus voltage, Vbus	135 V
Applied load impact	5 Nm
Hysteresis	3
Hysteresis deviation	0
Decimation	17

4.4. Open loop operation

Figure (4-2) shows the complete case study simulation test in continuous time.

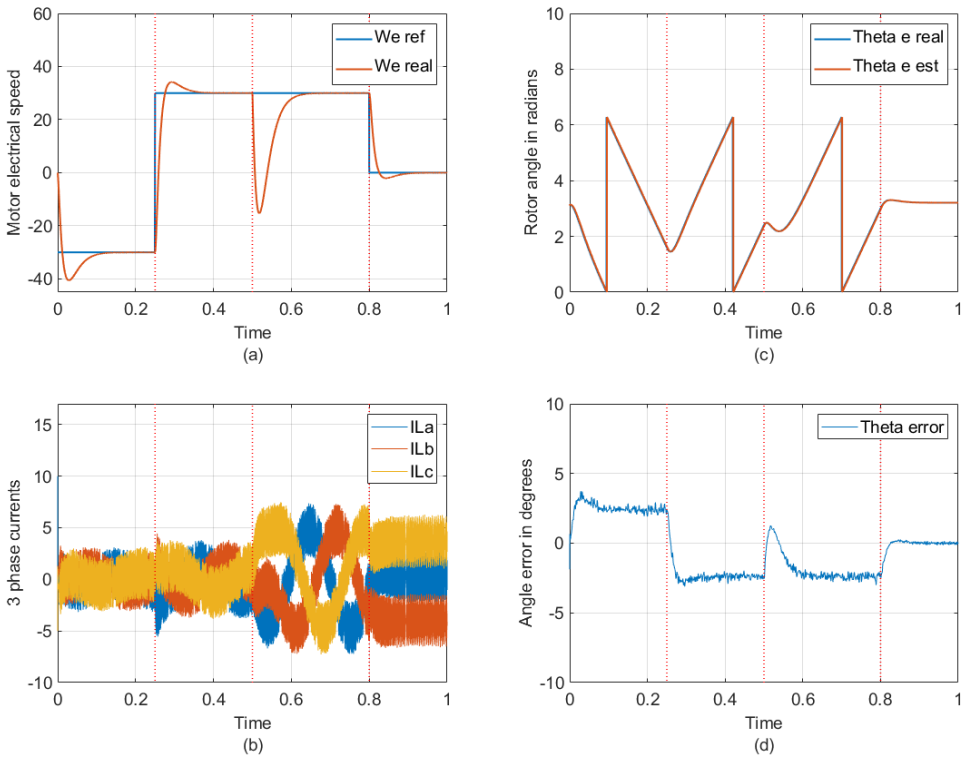


Figure 4-2: Case study in open loop continuous time

An ideal estimator working in open loop condition means that the estimated angle is not being injected into the system. It can be observed in the case study Figure (4-2) (c) that the estimated angle always tracks the real angle with some error Figure (4-2) (d).

4.4.1. Simulation test results and details

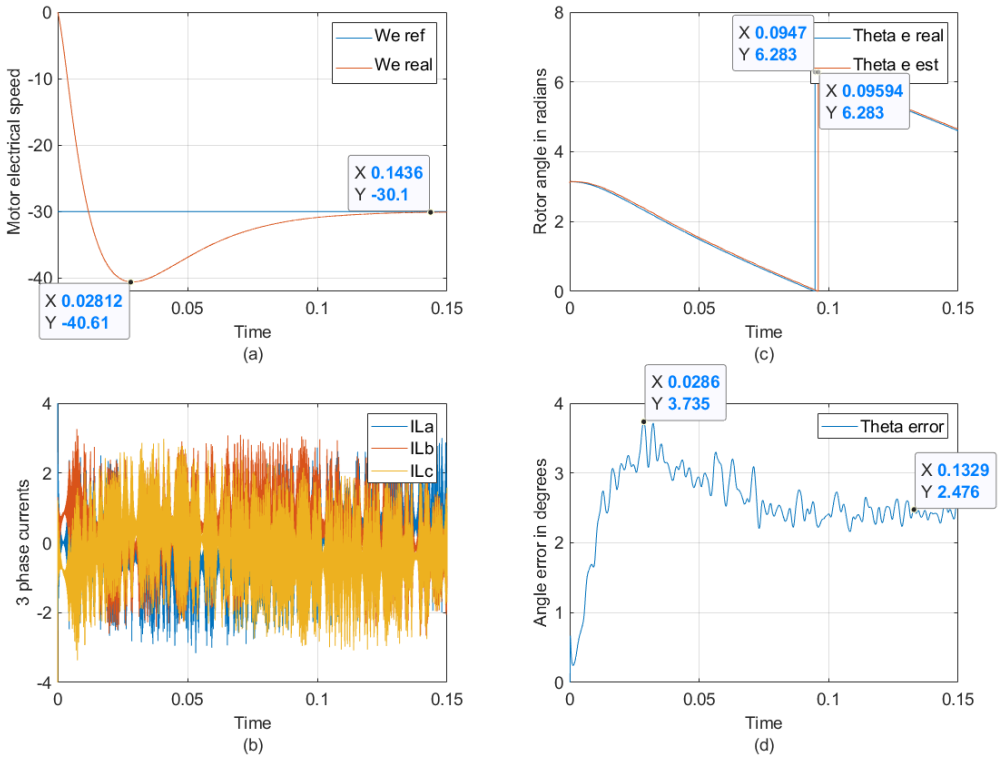


Figure 4-3: System start-up test in open loop continuous time

Figure (4-3); (a): Describes the system start-up with a settling time of 144 ms, (b): Motor currents have high frequency ripples and are not zero as motor is moving, (c): Estimated angle is tracking the real one, (d): In transient error approaches to a peak of 3.7 degrees and in steady state the average value of error is around 2.5 degrees which is quite good.

Figure (4-4); (a): A speed reversal can be observed with a settling time of 158 ms, (b): Movement in the motor currents can be observed, (c): Slope reversal in both angles, estimated and real ones verifies the speed reversal, (d): In transient the error reaches to -3.1 degrees and in average in steady state the error is around 2.3 degrees, more or less same as before.

Figure (4-5); (a): A transient in the real speed due to the load (value of 5 Nm) that is attached to the rotating motor, can be observed, which describes that the motor slows down, even stops at one point and rotates in the opposite direction for a moment, this transient disappears within a settling time of 180 ms system goes back to steady state, (b): A rise in currents, with a peak value of around 7 A, can be observed which verifies that a load has been attached with the machine (notice that the currents are synchronized and 120 degrees apart, as in a three phase balanced system), (c): A transient in the angles can be observed and observe how accurately the proposed estimator is estimating the angle, (d): In transient error reaches 1.2 degrees which is still reasonable and in steady state does back to -2.3 degrees.

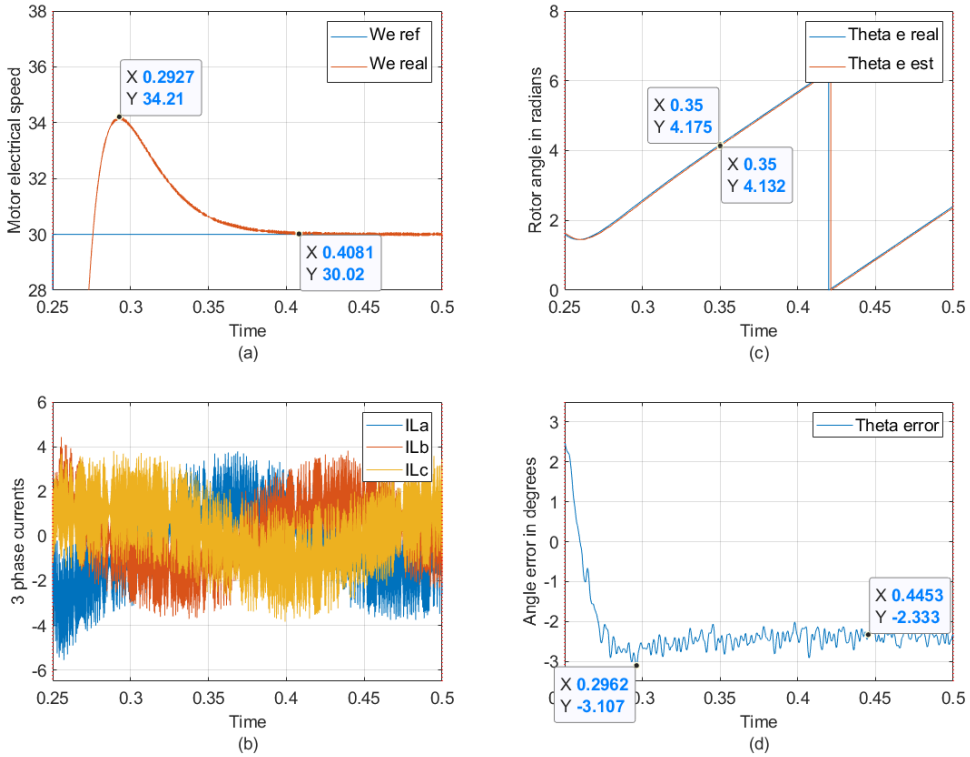


Figure 4-4: Speed reversal test in open loop continuous time

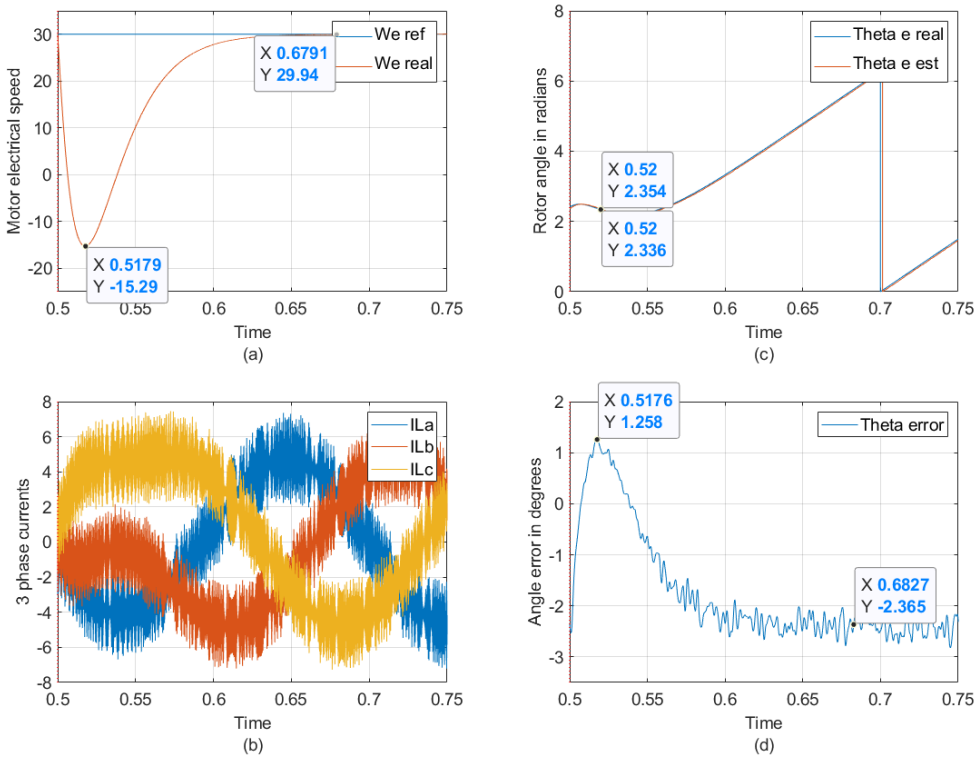


Figure 4-5: Load impact test in open loop continuous time

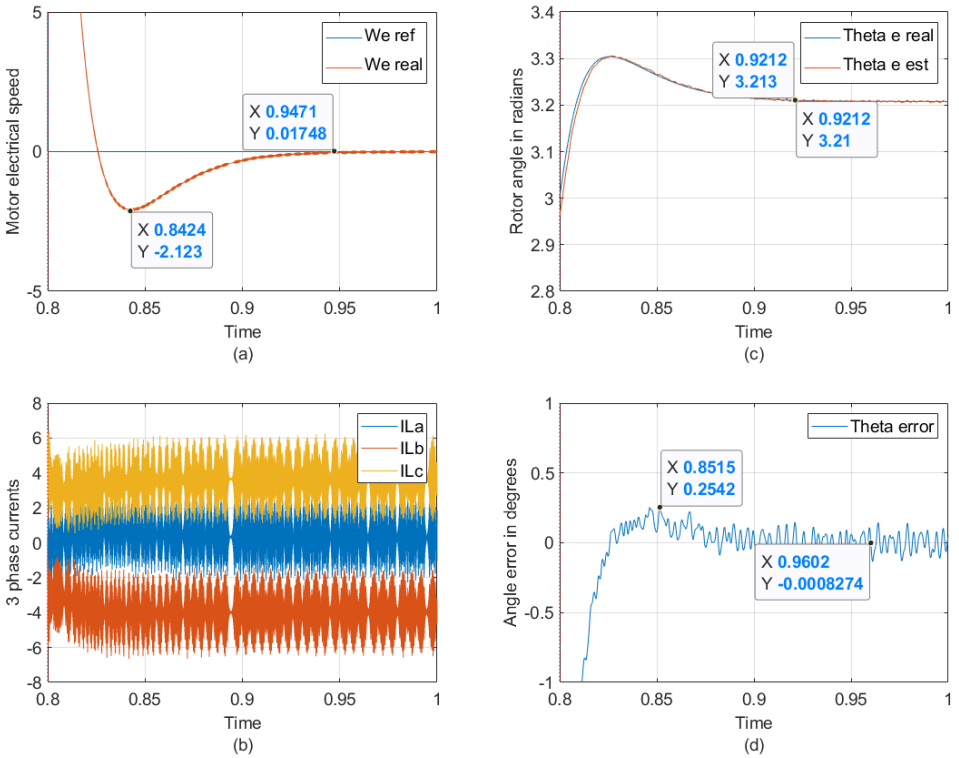


Figure 4-6: Zero speed test in open loop continuous time

Figure (4-6); (a) At this point the motor is stopped and the settling time is around 147 ms, which is more or less the same as measured before, (b): All the currents becomes constant and the sum of these must be zero which can be verified just by considering the mean value of them, (c): Angle estimation becomes a constant value and tracks the real theta as motor is stopped, (d): The mean value of the estimation error also goes zero.

Therefore, talking about the estimation in this last test proves the main objective of this estimation algorithm that is to estimate the angle even for a stopped motor, which also has been discussed before in this document. Hence there is no injection of any signals, thus no increment in the motor losses and/or decrement in the system efficiency and rotor position is estimated all the time.

4.4.2. Speed estimation

As the speed estimation is not intended to be the part of this work, however, during the testing phase speed estimation has also been extracted and tested just to compare it later with the results obtained from the real system.

Additionally, no filter has been used to clean the speed estimation signal. Nevertheless, the same case study as before has also been simulated for the speed estimation to make it consistent with rest of the simulations and to keep the transient response for the comparison later.

The graph in Figure (4-7) describes the speed estimation result in which, during the complete simulation, a ripple of ±25 - 30% of the nominal speed can be observed which is transients could reach up to ±60% of the nominal speed. From the speed estimation point of view is not too good and that may cause an unstable behaviour of the system if it is injected instead of the real speed.

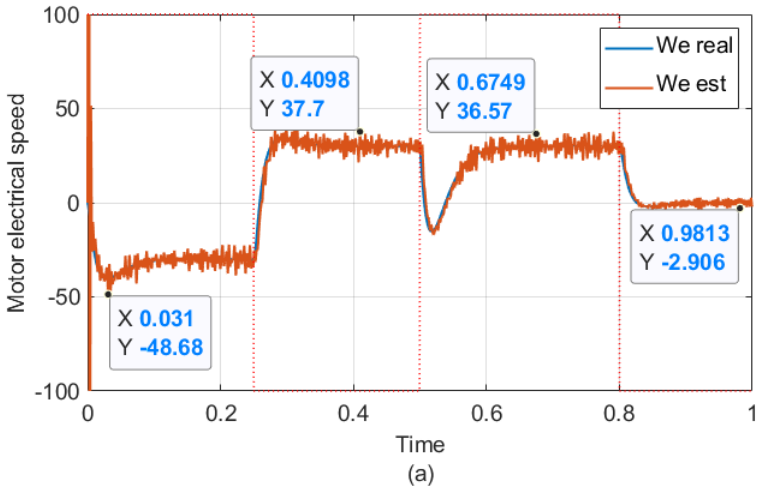


Figure 4-7: Speed estimation in open loop continuous time

As the estimation of the speed is not good enough, a lot of noise can be observed. Since in the theta scaling block, the method that is used to estimate the speed is taking the derivative of the estimated angle, from the following relation:

$$\omega_e = \frac{d\theta_e}{dt}$$

therefore, due to the variations in the theta estimation, the derivative tries to reach to its maximum value that can also be described as infinity, which results in very noisy output, in this case, the speed estimation signal.

4.5. Sensorless operation

In this section, a sensorless operation has been performed and the results have been compared with the open loop simulations.

Two types of tests are performed in this scenario:

1. Closing the loop once the speed is in steady state, around 0.1 s
2. Closing the loop before the speed reaches steady state, around 0.001 s

Results obtained in both cases have been shown in the following sections.

4.5.1. Sensorless activation once the speed is in steady state

In this part, basically we are going to close the loop in the system to perform a sensorless operation which implies injecting the estimated angle into system instead of real angle.

Figure (4-8) shows the complete case study simulation result of an ideal estimator working in closed loop condition, means that the estimated angle is being injected into the system at time 0.1 s.

It can be observed in the case study Figure (4-8) (c) that the estimated angle still always tracks the real angle with some error Figure (4-8) (d).

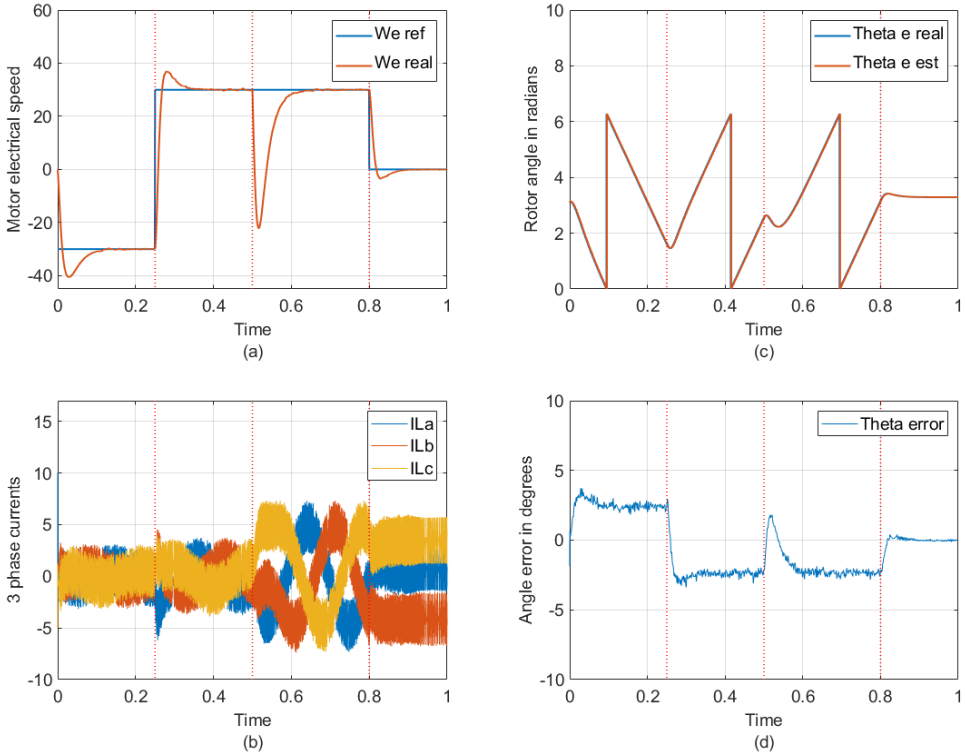


Figure 4-8: Case study in sensorless continuous time - I

4.5.1.1. Simulation test results and details

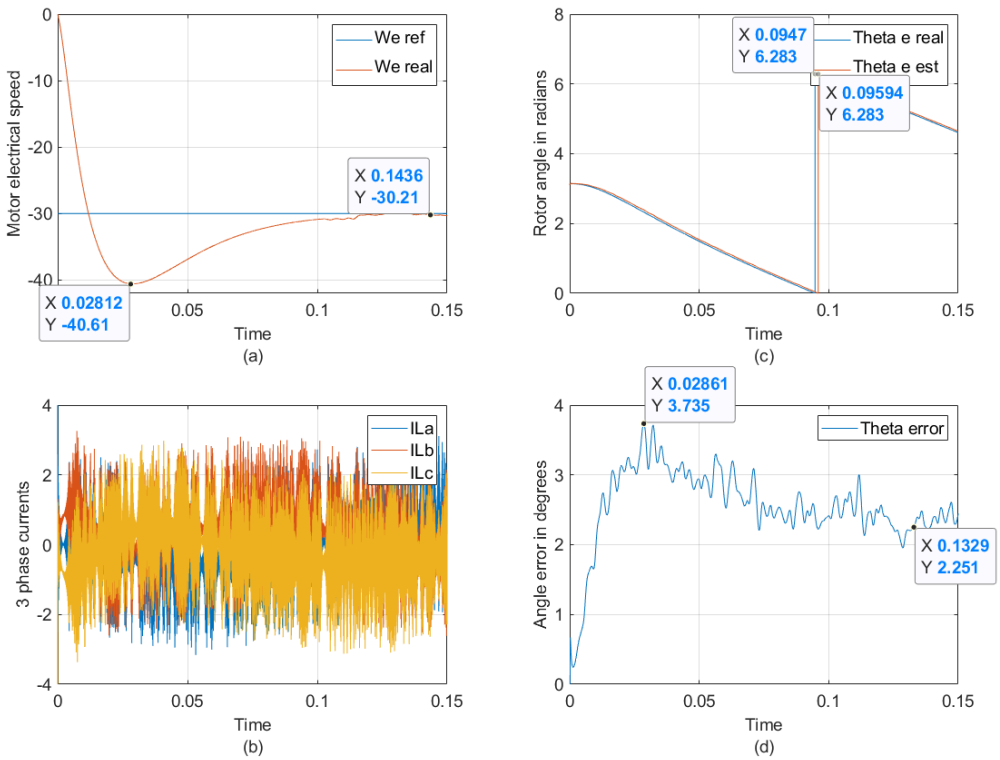


Figure 4-9: System start-up test in sensorless continuous time

Figure (4-9); (a): Describes the system start-up with a settling time of 144 ms, same as in open loop condition, (b): Motor currents looks the same as before with high frequency ripple,

(c): Estimated angle is following the real one, (d): As compared to Figure (4-3) (d), after 0.1 s a different error pattern can be observed because at this point we are injecting the theta with the error into the real system, in steady state the average error is around 2.3 degrees which still is a good value.

Figure (4-10); (a): As it can be observed that the real speed of the motor becomes noisy once the transient is passed this is due to that fact that there is noise in the estimated theta that is being injected into the system, settling time is again 158 ms, (b): Movement in the motor currents can be observed as before, (c): Slope reversal in both angles, estimated and real ones verifies the speed reversal, (d): In transient the error reaches a peak of -3.35 degrees (a bit more than the open loop test) and in average in steady state the error is around 2.3 degrees, more or less same as before.

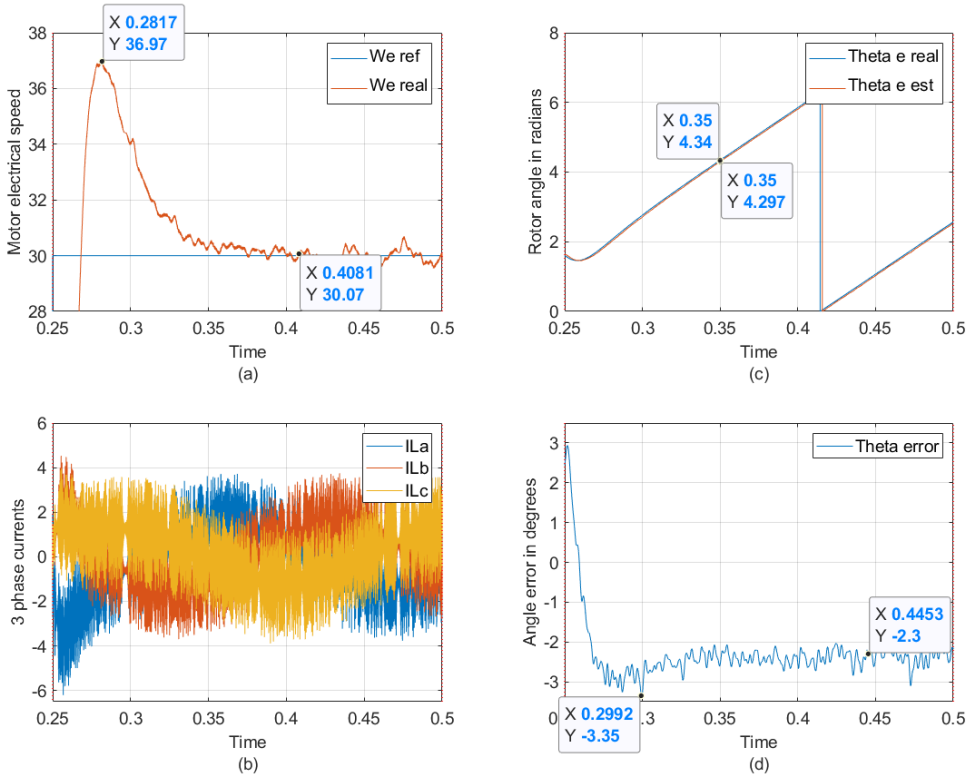


Figure 4-10: Speed reversal test in sensorless continuous time

Figure (4-11); (a): A transient in the real speed due to the load (value of 5 Nm) that is attached to the rotating motor, can be observed, which shows the same effect on speed as in Figure (4-5) (a), with same settling time of 180 ms the system achieves the steady state, (b): A rise in currents, with a peak value of around 7 A, can be observed which verifies that a load impact (again the currents are synchronized and 120 degrees apart, as in a three phase balanced system), (c): A transient in the angles can be observed but still the estimation accurately tracks the real motor angle, (d): Compared to Figure (4-5) (d), error is a bit bigger but it still converges and does to a mean value around -2.4 degrees.

Figure (4-12); (a) At this point the motor is stopped and the settling time is around 147 ms, (b): All the currents becomes constant but their pattern behaviour is different than Figure (4-6) (b) and still their sum is zero which can be verified just by considering the mean value of them, (c): Angle estimation becomes a constant value and tracks the real theta as motor is stopped, (d): The mean value of the estimation error also converges to zero after transient

but if it is compared with Figure (4-6) (d) the error pattern is totally different and small ripples in the error can be verified.

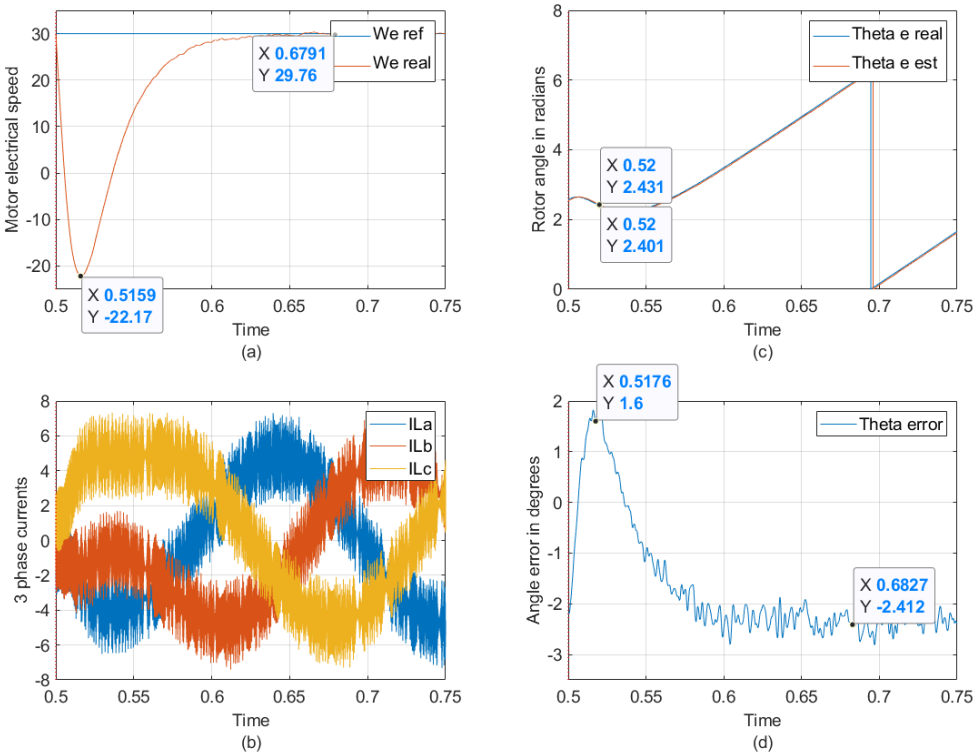


Figure 4-11: Load impact test in sensorless continuous time

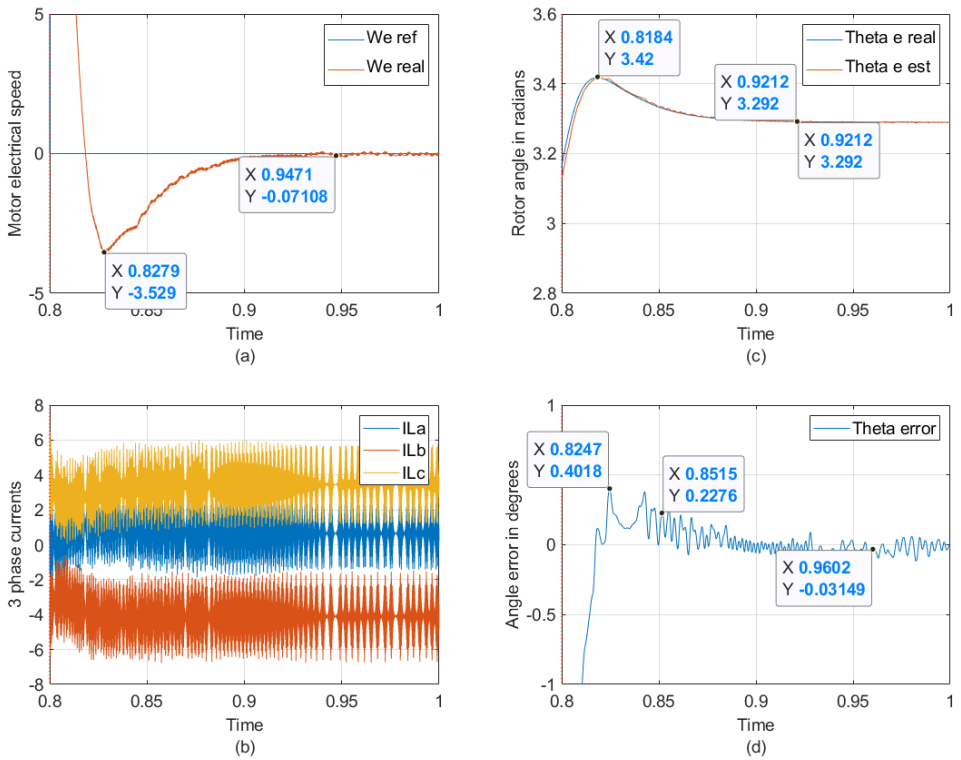


Figure 4-12: Zero speed test in sensorless continuous time

4.5.2. Sensorless activation before the speed reaches steady state

Same as before, Figure (4-13) shows the complete case study in closed loop condition and angle is being injected into the system at time 0.001 s:

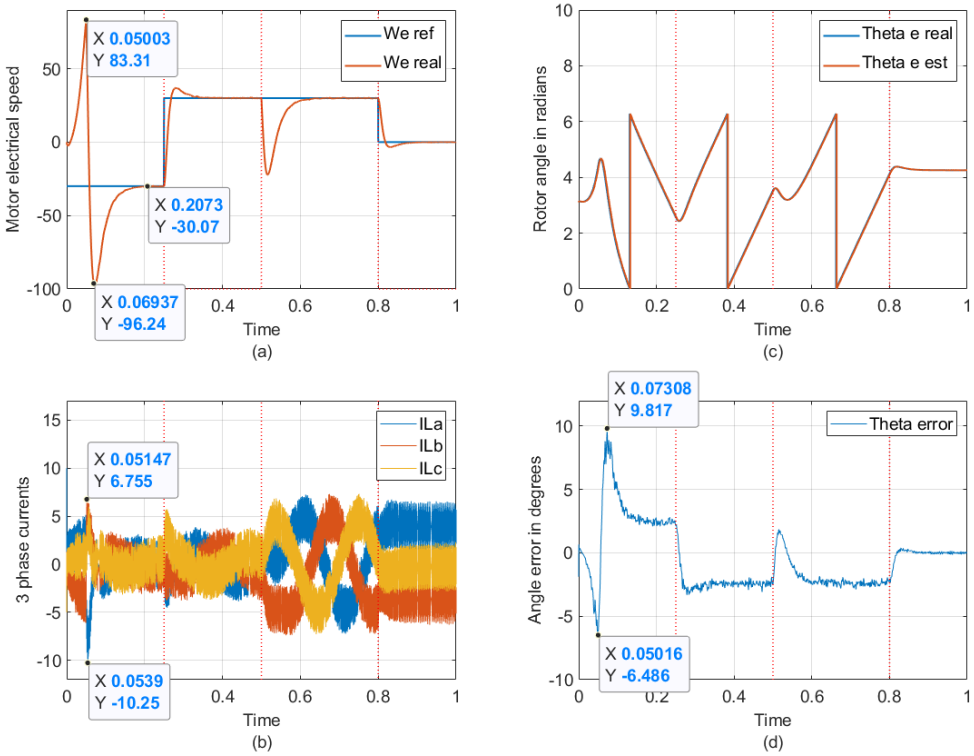


Figure 4-13: Case study in sensorless continuous time – II

Notice the behavior of the speed in the first transient, compared to Figure (4-8), here speed reaches to its maximum value, that happened due to the sensorless activation in the speed transient, and yet due to the high dynamics of the SMC, speed converges to the reference speed around 200 ms and afterwards the results are the same as described in Figures (4-8, 4-9, 4-10 and 4-11).

4.5.2.1. Speed estimation

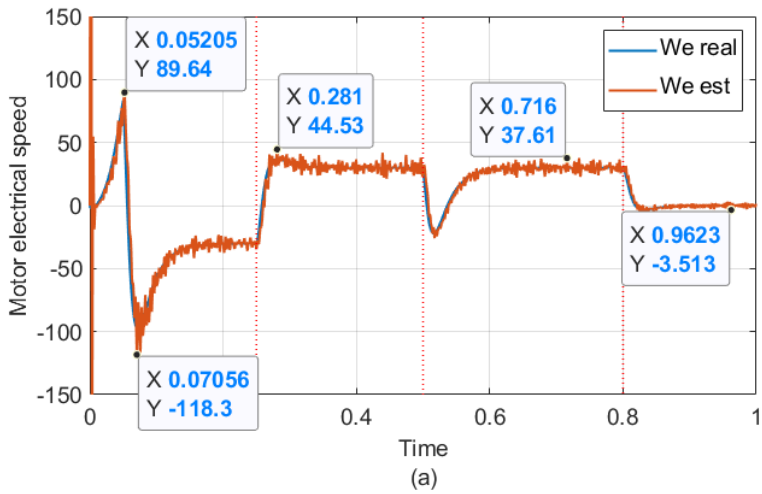


Figure 4-14: Speed estimation in sensorless continuous time

The graph in the Figure (4-14) describes the speed estimation in the sensorless operation of the system during the complete simulation, except for the first transient part where sensorless operation is activated during the speed transient, a ripple of ±25-60% of the nominal speed can be observed. Compared to Figure (4-7) this case is worse.

4.6. Position signals estimation

Figure (4-15) shows the position signals obtained in the estimation process for the same case study described before;

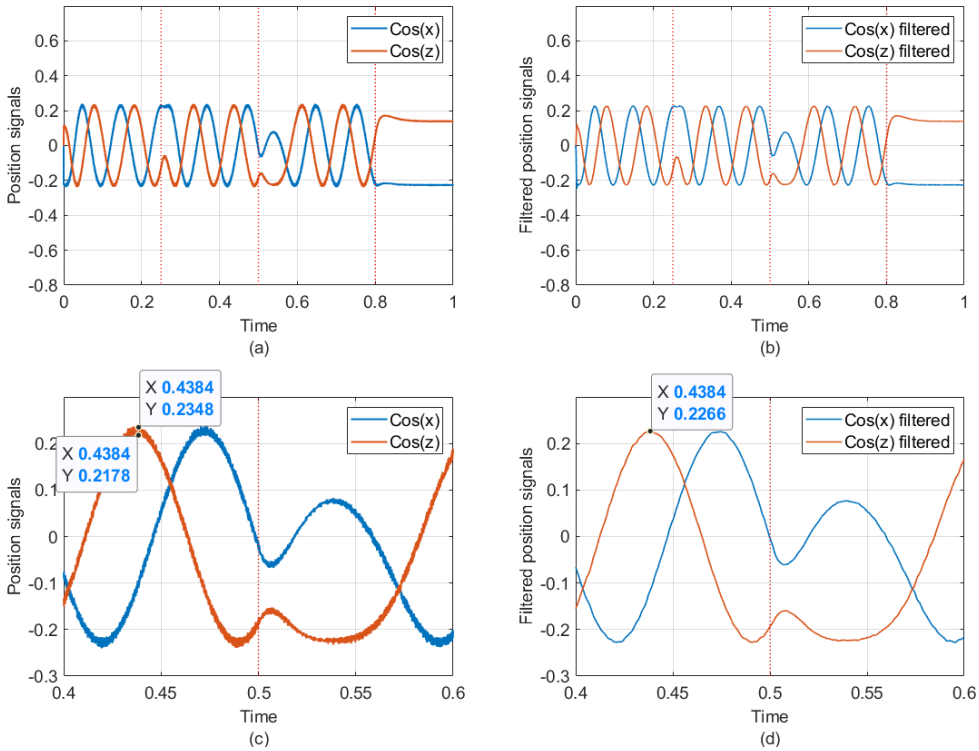


Figure 4-15: Position signals in open loop continuous time

Graph (b) is the filtered version of (a) and graphs (c) and (d) are zoomed versions around the load impact non-filtered and filtered respectively. These position signals appear with $2\theta_e$ information as shown before in the estimator block diagram, due to the fact that they are extracted from the saliency effect which depends on the salient poles of the motor.

4.7. Butterworth filter implementation

A Fourth-order Butterworth low-pass filter (LPF) is implemented as low-pass filter to clean the position signals. Following graphs in the Figure (4-16) shows the filter response.

Position signals are the main source of our information, thus, by removing the noise from the position signals, with as minimum as possible phase shift, a better angle estimation has been achieved.

It can be seen in the Figure (4-16) that the -3 dB cut-off frequency of the filter is around 300 Hz. However, the signals that we are trying to filter are mapped on very low frequency range, even lower than 20 Hz. Thus, the phase shift added by this filter to our signals is less than 5 degrees which is already verified in the angle error graphs that apart from the transient and if motor is moving the average phase error is between 2-3 degrees only.

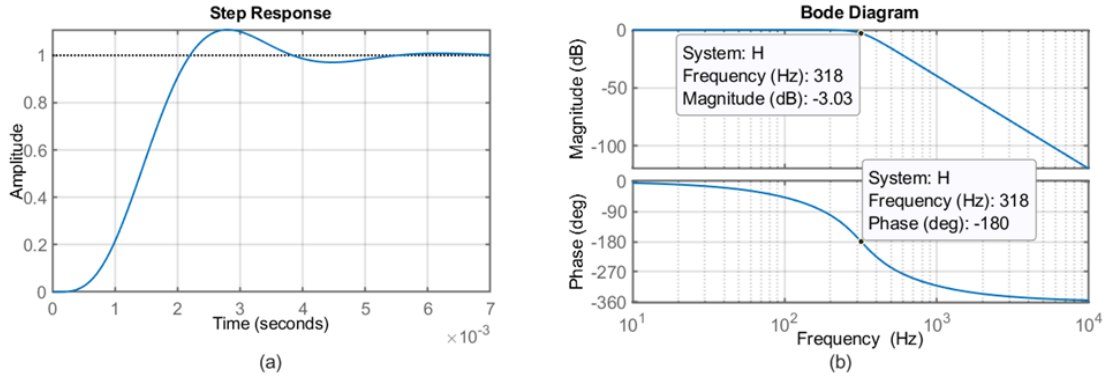


Figure 4-16: Butterworth filter response

Following expressions demonstrate the transfer function of this filter:

$$\mathrm{TF}_{\mathrm{Butterworth\_filter}} = \frac{1}{6.2510^{-14}s^4 + 3.26610^{-10}s^3 + 8.53610^{-7}s^2 + 1.30710^{-3}s + 1}$$

An FFT analysis is also performed in the real system, which then is verified by the theoretical calculations, that gives us the exact information regarding the frequency of the position signals. Following Table 6 describes the relation between the motor speed and position signals frequency that appears during the FFT analysis:

Table 6: Relation between motor speed and $2\theta_e$ signals

Electrical speed $\pm\omega_e$ (rad/s)	Calculated signal frequency in θ_e (Hz)	Measured signal frequency in $2\theta_e$ (Hz)
0	0	0
5	0.795	1.6
10	1.59	3.2
15	2.38	4.8
20	3.18	6.4
25	3.97	8.0
30	4.77	9.6

Where the calculated signal frequency values are obtained with the direct relation of the motor speed and frequency by using the following relation:

$$f = \frac{\omega_e}{2\pi}$$

Therefore, measured signal values are 2 times of the calculated values and that make totally sense because the above frequency speed relation is described in θ_e only, where the measured signals from the real system are obtained in $2\theta_e$ information.

Hence a limit for the speed has to be defined in order to design a low-pass or band-pass filter to clean the position signals and obtain clean estimation, thus, in our design, the worst case speed has been defined as ±30 rad/s.

5. Algorithm discretization

After completing the continuous time simulations and concluding test results, now it is time to discretize the system and perform the same tests before switching to the real system. Therefore, discretization of the algorithm has been performed and the same case study as before has been simulated. Firstly, to make a comparison between continuous time and discrete time simulations and secondly to keep the outputs results as close as possible to the results that would obtain later in the experimental system.

Therefore, while performing the discretization process, following amendments has been made into the system:

1. In a digital signal processing system (DSP) system, the code executes in routines with priorities; hence algorithm is divided into two sub routines:
    a. Fast execution routine
    b. Slow execution routine
2. Derivative of the slopes for estimation are executed in the fast routine.
3. The rest of the estimation algorithm is executed in the slow routine.
4. Sliding mode current control (SMCC) is being executed in the fast routine.
5. A PWM is implemented in the system which does not work as a conventional PWM but as a set reset flip-flop in order to implement the hysteresis comparator [30].
6. Dead times are added into the control signals used for VSI.
7. Low-pass filters are added in the current measurements whose cut off frequency is around 1 MHz.
8. A typical 12-bit resolution is taking into the consideration for the currents and other signals as like of an ADC in real system.

Note II: *More information regarding the algorithm subroutines are provided in the following Chapter "Implementation and experimental results".*

A very common effect of the discretization is that; it introduces more noise into the signals due to the sampling effect that can be observed in the results presented later in this Chapter.

5.1. Simulation data

The simulation parameters in discrete time have been changed and adopted to get more realistic values.

Note III: *These simulations have been run twice, first the system was designed for 5 µs execution time for fast routine and 125 µs execution time for slow routine. Later after testing it in DSP, we found that there was not enough time for the system to run. Firstly, because the DSP already had a big amount of code to execute for the SMC and secondly the DSP model that is used is too old, so it has its own limitations. Due to that, there was not enough time to execute the estimator for desired times, thus, after performing the initial tests in the real system, the execution times for the subroutines was changed and discrete time simulations were run again with the new execution times to make them as realistic as possible. The new timings are mentioned in the Table 7 below.*

Additionally, the speed controller has different gains now as compared to continuous time simulations because it had been designed already by the author of the prerequisite work of the proposed estimator, hence, the output results are different as described before and the transients in the speed are much more of a critically damped response.

Table 7 contains the new parameters of the discrete time simulations:

Table 7: Simulation parameters for discrete time

Parameter	Value
Simulation step time, T_s	6.667 ns
Simulation time	1 s
Speed controller proportional gain, k_p	0.3
Speed controller integral gain, k_i	0.0006
Rotor initial position	π = 3.1415
Bus voltage, V_{bus}	135 V
Applied load impact	5 Nm
Sampling time, T_{samp}	6 µs
Fast routine execution time	6 µs
Slow routine execution time	150 µs
Hysteresis	1.8
Hysteresis deviation*	20%
Decimation	10000

"*" Explanation for the hysteresis deviation is provided in the topic "5.2".

Therefore, the fast routine is being executed at 6 µs and slow routine at 150 µs ensuring that the DSP has enough computation time to run the complete control system and rotor angle estimator along with it.

In the beginning of the discrete time simulations, a phenomenon in the control signals was observed, which is; the control signals start synchronizing most of the time. And if the control signals are synchronized, the angle estimator cannot perform the estimation. However, this problem was not encountered during the continuous time simulations.

The following topic explains this issue and its solution with more details and graphs.

5.2. Control signals synchronization phenomenon

During discrete simulations we found that the control signals synchronization happens more often than the continuous time simulations. The control actions are generated according to

[30]. So due to the synchronization, it was not possible to detect the saliency effect in the sigma signals most of the time because when the controls were synchronized and two control actions; all zeros and all ones occurs more often. According to the Table 3, for these two control actions, there is no information for the position signals thus, no angle estimation. Figure (5-1) shows the synchronized control signals and sigma signals behaviour.

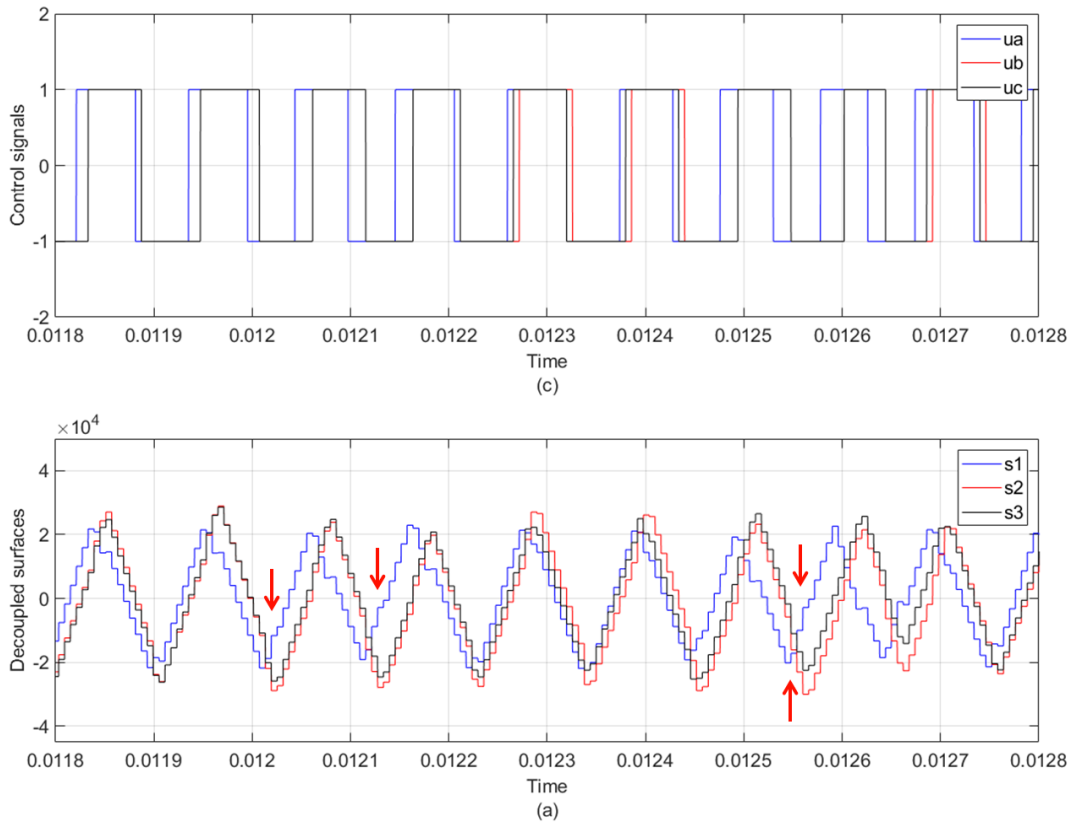


Figure 5-1: Synchronized control signals and sigma surfaces

The sampling time in this case is 6 µs as mentioned in the parameters of discrete time simulation section. The red arrows indicate the change of slopes in sigma signals. The exact cause of this control signals synchronization is not been identified yet.

Nevertheless, to desynchronize the control signals following two solutions have been proposed and verified.

1. In fixed hysteresis control just by alleviating the hysteresis value with 20% the control signals can be desynchronized. Therefore, one hysteresis control keeps the nominal hysteresis value and in the other two; 20% increment and reduction are applied in the hysteresis value.
2. In variable hysteresis control by performing a 20% variation in the switching period reference the control signals can be desynchronized. Therefore, one hysteresis control keeps the nominal switching period and in the other two; 20% increment and reduction are applied in the switching period.

The obtained results in both cases are similar, thus only one result is taken that can be observed in the Figure (5-2).

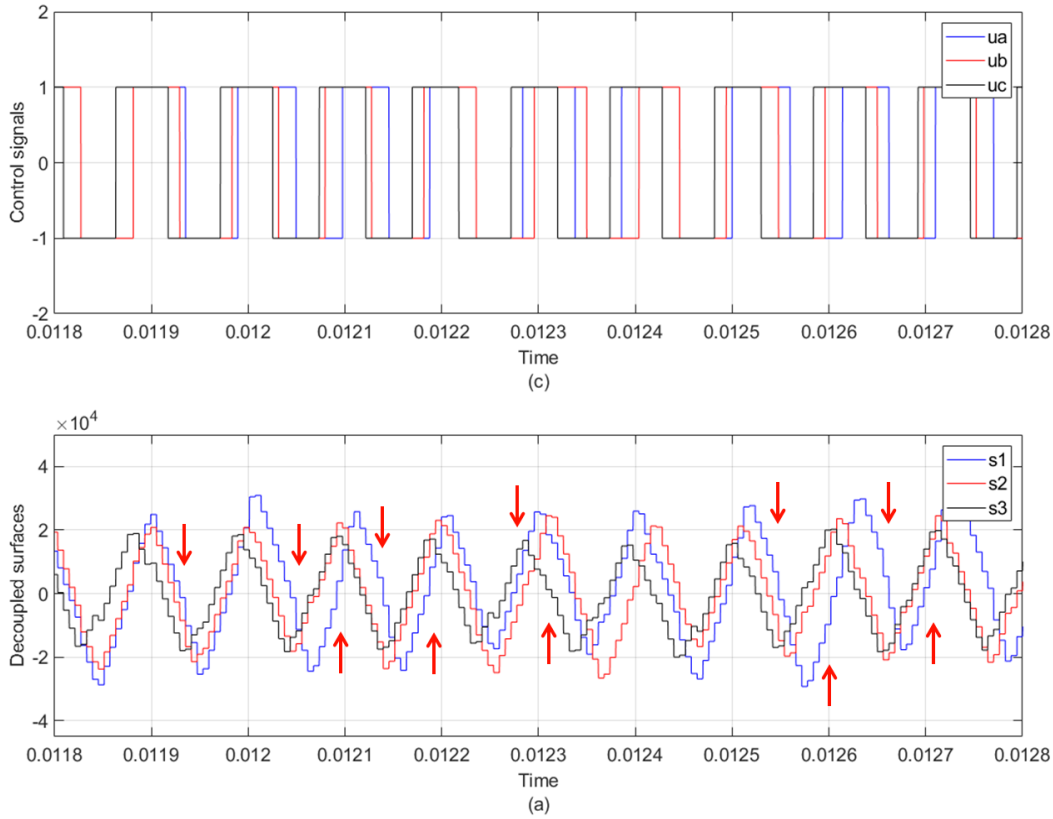


Figure 5-2: Desynchronized control signals and sigma surfaces

Again, red arrows indicate the change of slopes in sigma signals due to saliency effect and it can be easily visualized that after desynchronizes the signals by applying 20% variation in hysteresis values, change of slopes can be detected much more time than before as shown in Figure (5-1).

Moreover, as we know that in a SMC, hysteresis comparators generate the control actions; either with fixed hysteresis control or variable hysteresis control. Therefore, the control actions are being generated randomly by hysteresis comparators. Thus, due to that, it could be the reason for control signals synchronization. However more study needs to be done for the proper identification of the root cause of this synchronization.

On the other hand, if that assumption is true, it makes this issue to be an inherent problem for the proposed estimation algorithm. It could also happen in the continuous time simulation which in our case was never detected. However, the provided solution can overcome this problem that is also verified.

5.3. Open loop operation

The Figure (5-3) on the next page shows the complete case study behaviour of the estimator in open loop condition where the estimated angle is not being injected into the system.

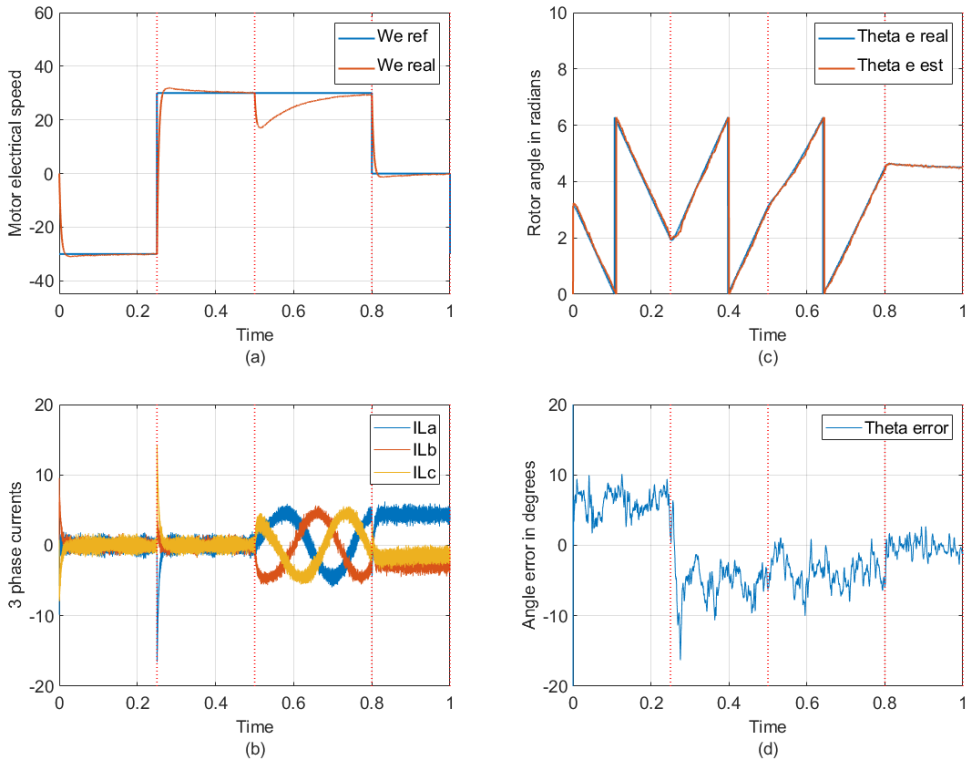


Figure 5-3: Case study in open loop discrete time

Angle estimation can be observed in graph (c) with an error shown in graph (d).

5.3.1. Simulation test results and details

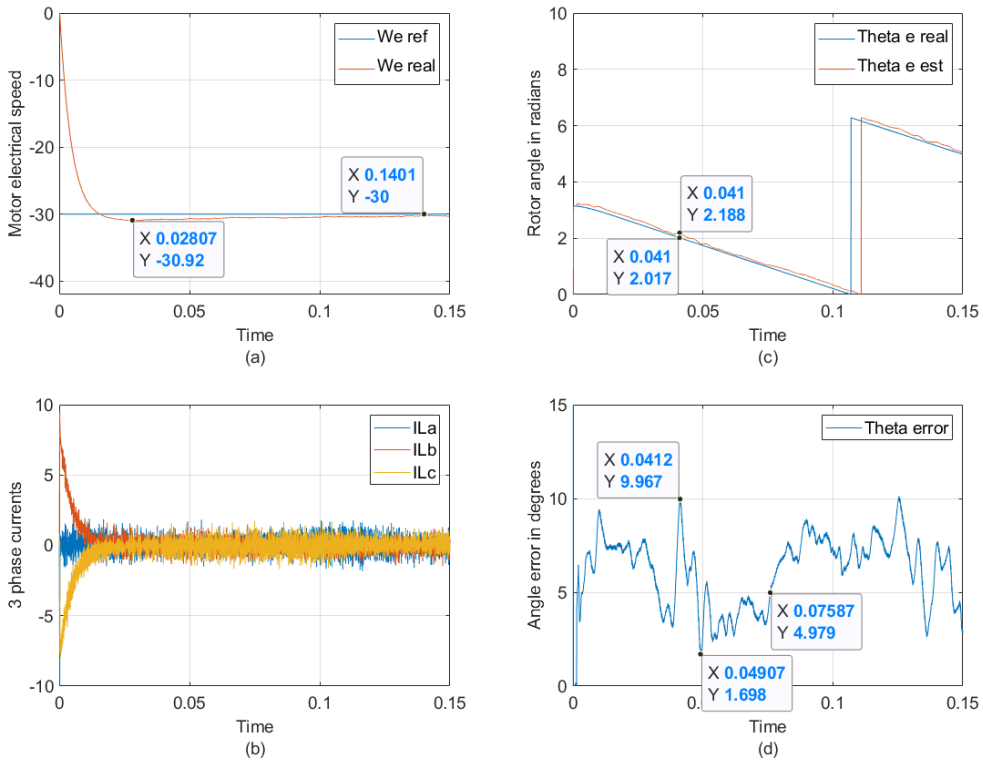


Figure 5-4: System start-up test in open loop discrete time

Figure (5-4); (a): Describes the system start-up with a settling time of 140 ms (overshoot is better as compared to continuous time simulation), (b): Motor currents have more high frequency ripples, (c): Estimated angle is tracking the real one but there is more phase shift, (d): Error is much more noisy than before, in transient error approaches to a peak of 10 degrees and in steady state the average value of error is around 5 degrees.

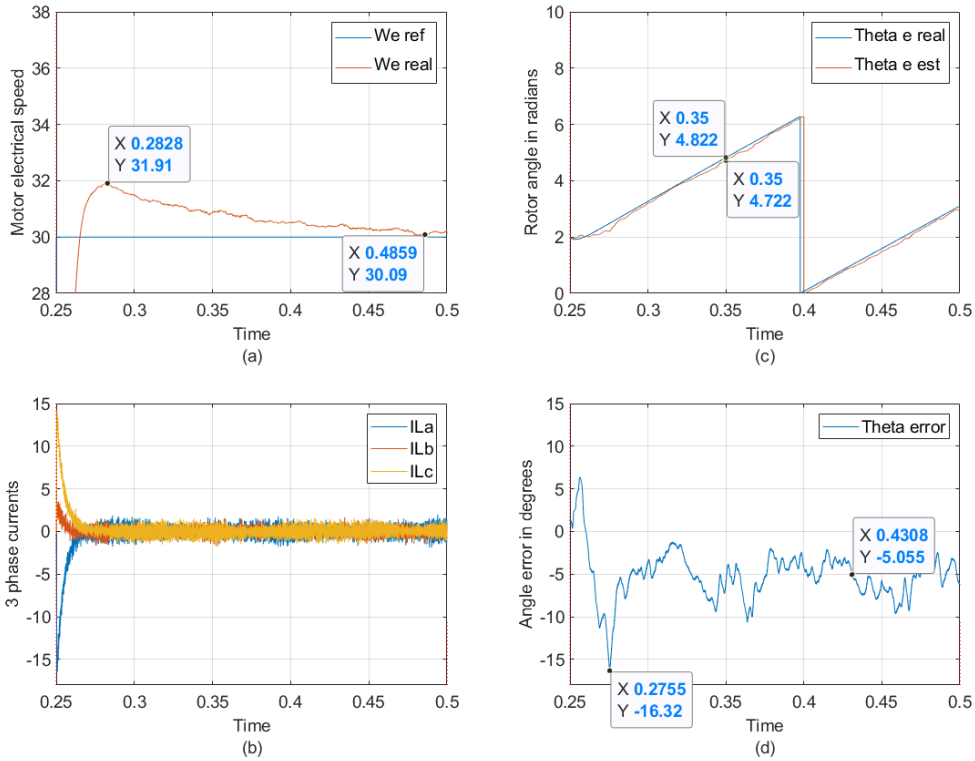


Figure 5-5: Speed reversal test in open loop discrete time

Figure (5-5); (a): A speed reversal can be observed with a settling time of 235 ms which is greater than before, but the overshoot in the transient is improved, (b): A transient in the motor currents can be observed, (c): Slope reversal in both angles, estimated and real ones, verifies the speed reversal, (d): In transient the error reaches -16 degrees and in average in steady state the error is around -5 degrees, more or less same as in start-up steady state but in negative values.

Figure (5-6); (a): A transient in the real speed due to the load impact (of value 5 Nm) can be observed, but with the new controller the motor does not rotate in opposite direction; the settling time for the transient is of 290 ms longer than continuous time, (b): A rise in currents, with a peak value of around 5-7 A can be observed, which verifies that a load has been attached with the motor and the currents are synchronized and 120 degrees apart, as in a three phase balanced system, (c): A transient in the angles can be observed and algorithm continuously estimates theta, (d): In transient the error stays between 1 and -10 degrees, which is reasonable and in steady state it goes back to around -5 degrees.

Figure (5-7); (a) At this point the motor is stopped and the settling time is around 190 ms, which is more as compared to the continuous time simulation, (b): All the currents becomes constant and the sum of these must be zero, (c): Angle estimation becomes a constant value and tracks the real theta as motor is stopped, (d): The mean value of the estimation error also converges to zero, but this time there is more ripple in the error which in some cases goes up to 3 degrees value.

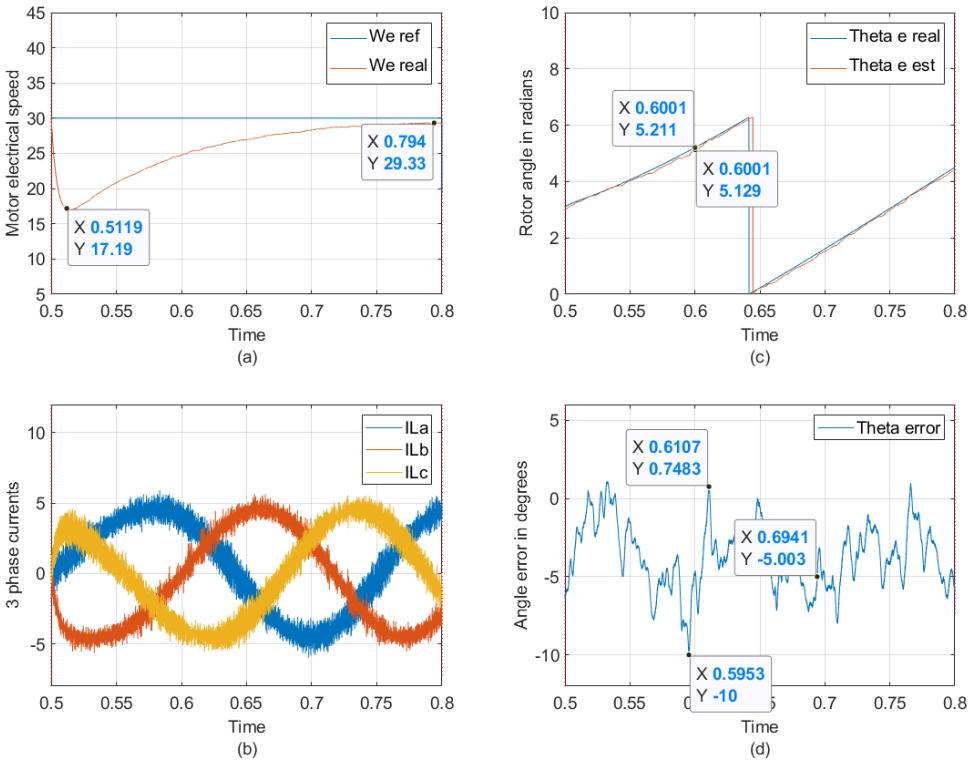


Figure 5-6: Load impact test in open loop discrete time

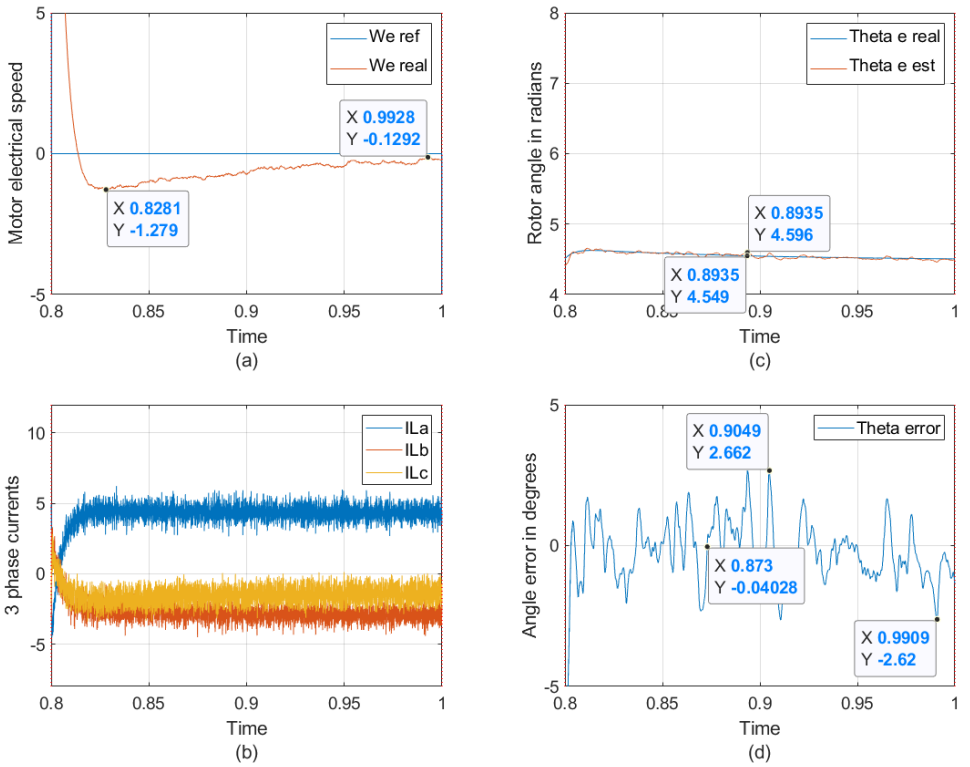


Figure 5-7: Zero speed test in open loop discrete time

Therefore, again, the estimation in this last test proves the main objective of this work that is to estimate the angle even for a stopped motor. Hence there is no injection of any signals,

thus no increment in the motor losses and/or decrement in the system efficiency and rotor position is always estimated.

5.3.2. Speed estimation

Figure (5-8) shows the speed estimation in discrete case.

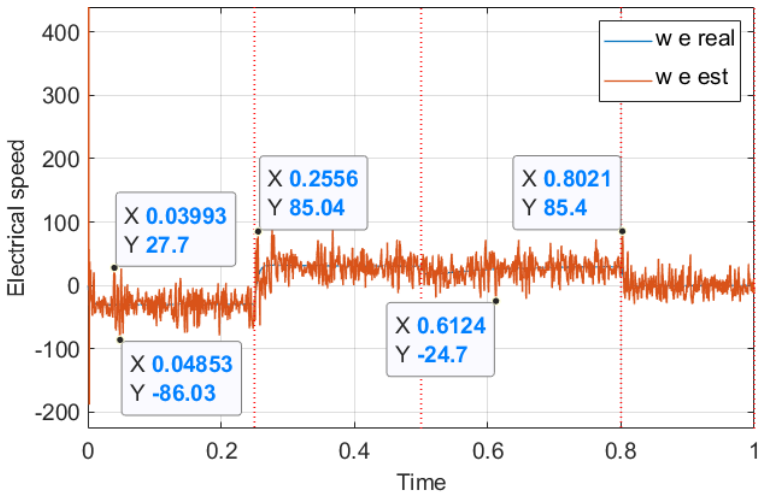


Figure 5-8: Speed estimation in open loop discrete time

Again, speed estimation is not intended but during simulation phase; speed estimation has also been extracted for all the cases just to compare it later with the results obtained from the real system.

Again, no filter has been used to clean the speed estimation signal. Nevertheless, the same case study as mentioned before has also been simulated for the speed estimation to make it consistent with all the performed simulations.

The graph in the Figure (5-8) describes the result in which during the complete simulation, a ripple up to ±186% of the nominal value of the speed can be observed including the transients. From the speed estimation point of view this is much worse than before (continuous time simulations). However, it is as expected because of discretization of the system. Remember no filter is being used to attenuate the noise from speed estimation.

5.4. Sensorless operation

For this section only one sensorless operation is performed which is when the real speed is in already reached the steady state, around 0.1 s.

The simulation result obtained is shown in the Figure (5-9) shows the complete case study behaviour of the estimator working in closed loop condition in discrete time which still is good enough.

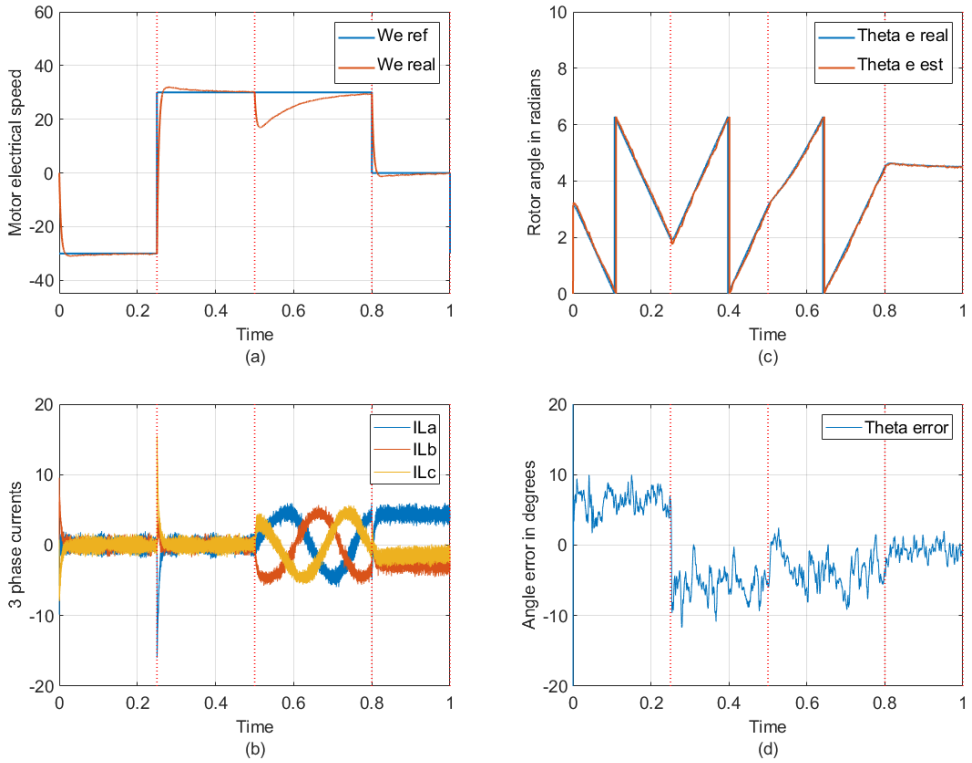


Figure 5-9: Case study in sensorless discrete time

Figure (5-9); (c): The estimated angle still always tracks the real angle with error and in (d): Angle error is different and obviously a bit more than before.

5.4.1. Simulation test results and details

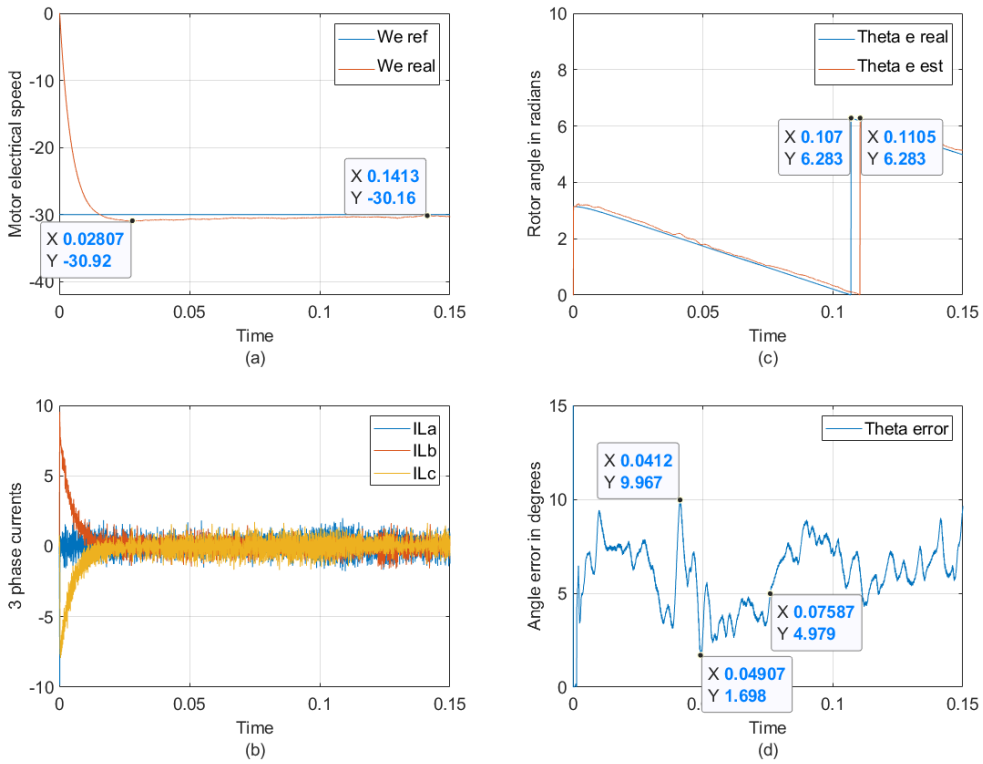


Figure 5-10: System start-up test in sensorless discrete time

Figure (5-10); (a): System start-up is still good with a settling time of 141 ms (b): A high frequency ripple in the motor currents can be observed, (c): Angle estimation is becoming noisy but still tracks the real one, (d): As compared to Figure (5-4) (d), after 0.1 s a different error pattern can be observed, in steady state the average error is between 6-8 degrees which still is a good enough as long as the error is not diverging continuously.

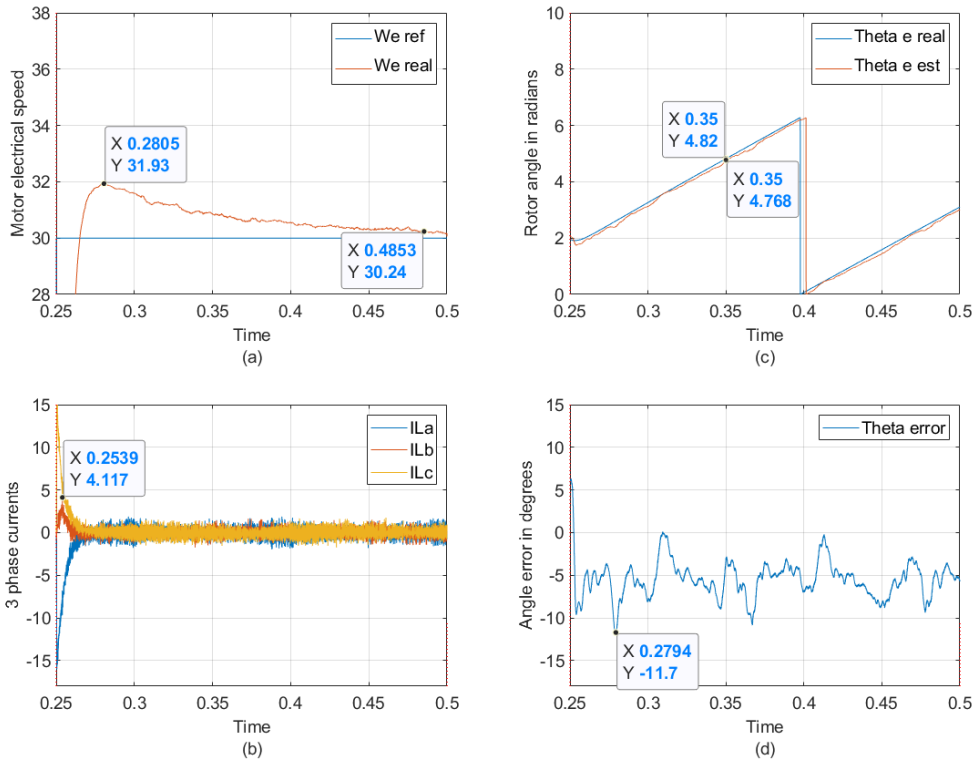


Figure 5-11: Speed reversal test in sensorless discrete time

Figure (5-11); (a): After the transient the speed becomes noisy due to that fact that there is noise in the estimated theta that is being injected into the system and the settling time is 235 ms, (b): There are small currents but with noticeable ripple, (c): Slope reversal verifies the speed reversal, (d): In transient error goes to -11 degrees (a bit more than the open loop test) and in steady state the mean error around 5-6 degrees with a variation of ±5 degrees.

Figure (5-12); (a): A transient in the real speed due to the load impact (of value 5 Nm) can be observed, which shows the same effect on speed as in Figure (5-6) (a), with a settling time of 300 ms, (b): A rise in currents, with a peak value of around 5-7 A, (c): A change of slopes in the angles can be observed but still the estimation tracks the real motor angle, (d): Compared to Figure (5-6) (d), error is a bit bigger but it still converges and does to a mean value around -5 degrees more or less.

Figure (5-13); (a) Motor is stopped at 0.8 s and the settling time is around 200 ms, (b): All the currents becomes constant as expected and their mean sum is around zero, (c): Angle estimation becomes a constant value and tracks the real theta as motor is stopped, (d): The mean value of the estimation error also reaches to zero after transient but if compared with Figure (5-7) (d) the error pattern is totally different with peak to peak ripples of 4 to 5 degrees.

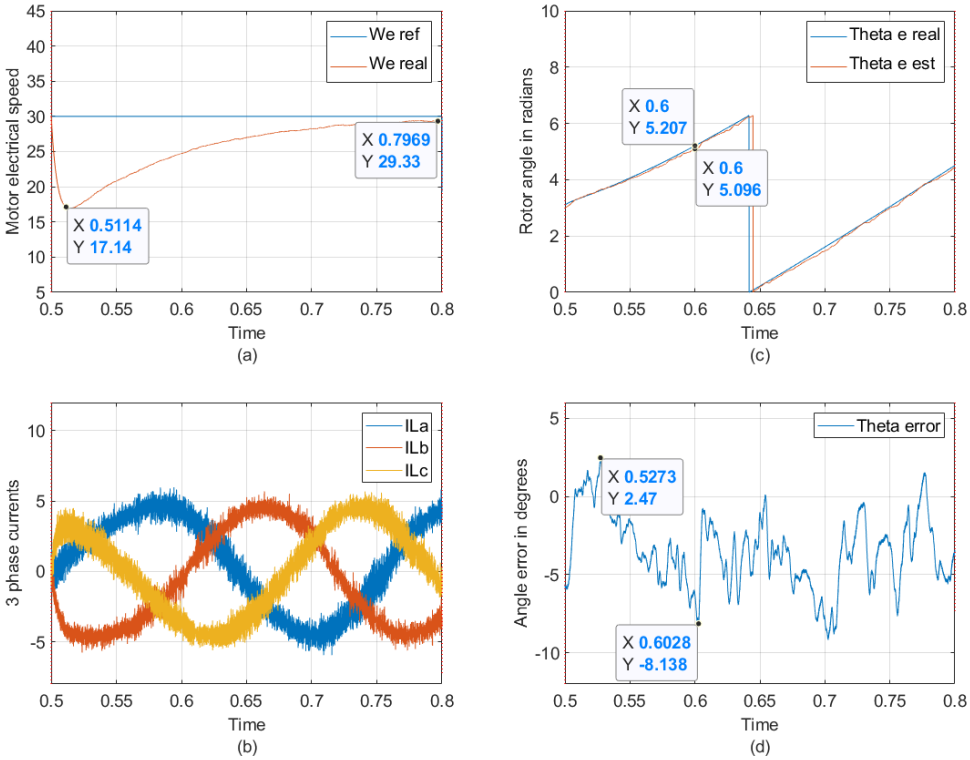


Figure 5-12: Load impact test in sensorless discrete time

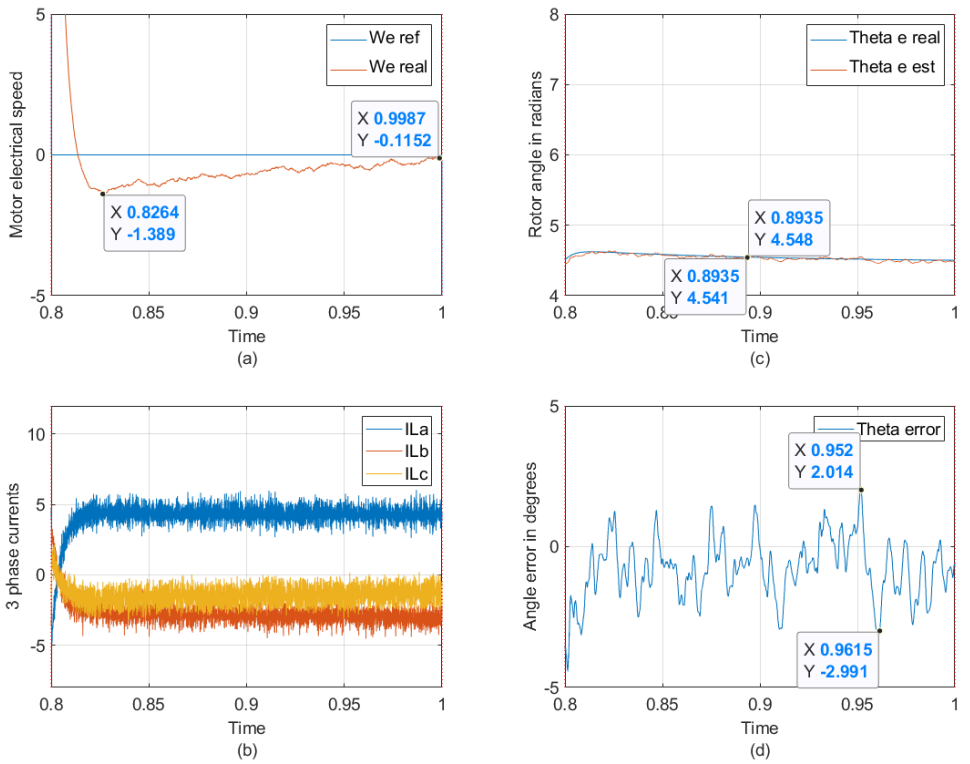


Figure 5-13: Zero speed test in sensorless discrete time

5.4.2. Speed estimation

Graph in the Figure (5-14) shows the speed estimation for sensorless case in discrete time simulation.

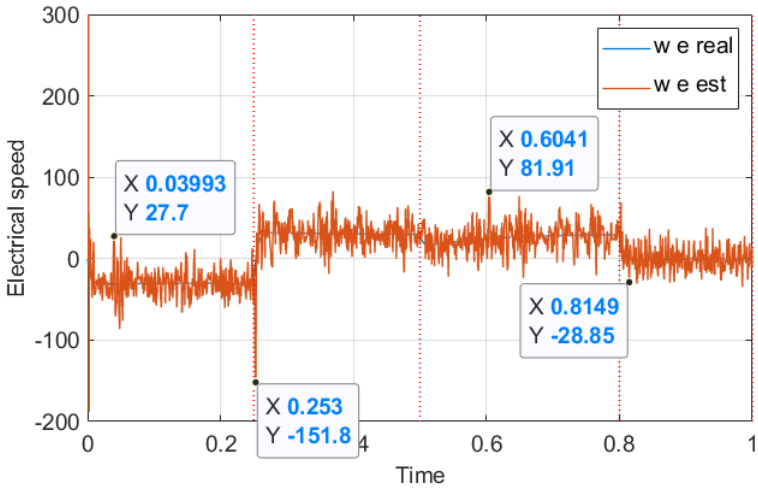


Figure 5-14: Speed estimation in sensorless discrete time

where the ripple is up to ±186% of the nominal speed value of same as in Figure (5-8) and even on speed reversal, at time 0.25 s, a transient up to -151 rad/s can be seen, which is from the speed estimation point of view is not good at all.

5.5. Position signals estimation

Figure (5-15) shows the position signals obtained in discrete time for the same case study:

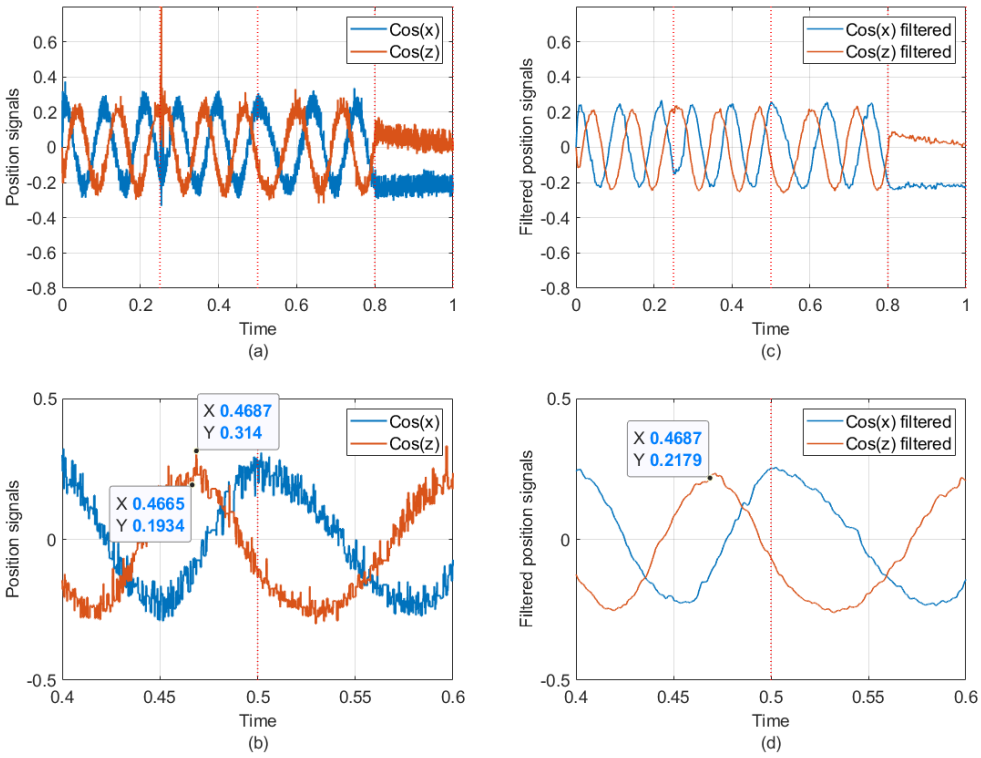


Figure 5-15: Position signals in open loop discrete time

Where, graph (c) is the filtered version of (a) and graphs (b) and (d) are zoomed versions around the load impact non-filtered and filtered respectively. These position signals appear with $2\theta_e$ information as shown before in the estimator block diagram, since they are extracted from the saliency affect which depends on the salient poles of the motor.

Notice that in this case, the signals are much noisier than the continuous time simulations. Of course, this is the effect of discretization and the error seen during the discrete simulations are coming from position signals with some phase shift added by the discrete time digital filter that has been used to clean these signals.

More details about the filter is described in the following topic.

5.6. 2nd order digital low-pass filter implementation

In this case a very simple 2nd order digital filter has been implemented and used to clean the position signals.

Note IV: *Considering that later in the real system the same filter or a very similar filter must be implemented.*

Hence a 2nd order digital low-pass filter has been implemented and whose filter response can be seen in the Figure (5-16).

Notice that this filter is adding more phase shift to the signals than the Butterworth filter that was implemented in continuous time.

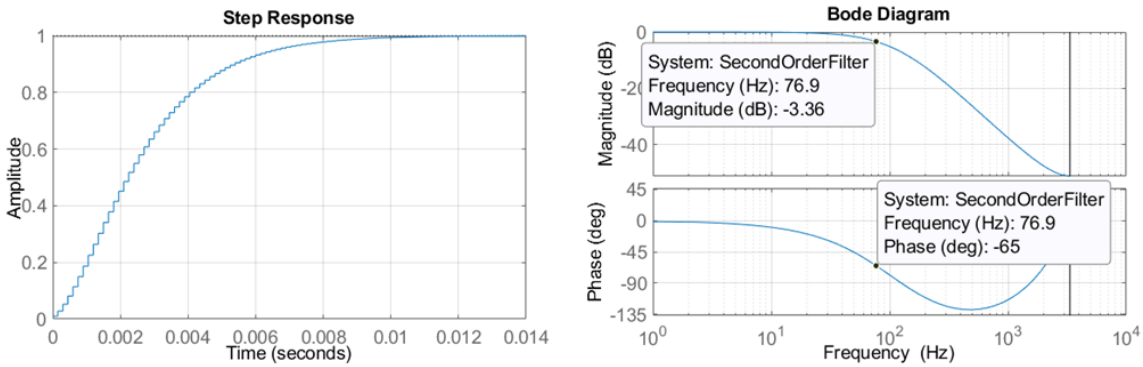


Figure 5-16: 2nd-Order digital filter response

As it can be seen, the -3dB cut-off frequency of the filter is around 75 Hz. However, the signals that we are trying to filter are mapped on very low frequency range, even lower than 20 Hz. But as this is a lower order filter as compared to Butterworth filter, hence the phase shift added by this filter to our signals is around 10 degrees for the worst case scenario which is already verified in the angle error graphs that apart from the transient and if motor is moving the average phase error is between 5-8 degrees.

Following expressions demonstrate the transfer function of this filter:

$$\mathrm{TF}_{\mathrm{2nd\_order\_digital\_filter}} = \frac{0.01z^2}{z^2 - 1.8z + 0.81}$$

where; the sampling time is 150 µs, because the part of the algorithm in which this filter has been implemented is in the slow execution routine.

6. Control system

This Chapter describes the complete control system and the rest of the system parts (that are not mentioned before) with all the interconnections and the flow of all the important signals.

6.1. Control system description and block diagram

Figure (6-1) depicts the block diagram of the control system and all the important associated signals.

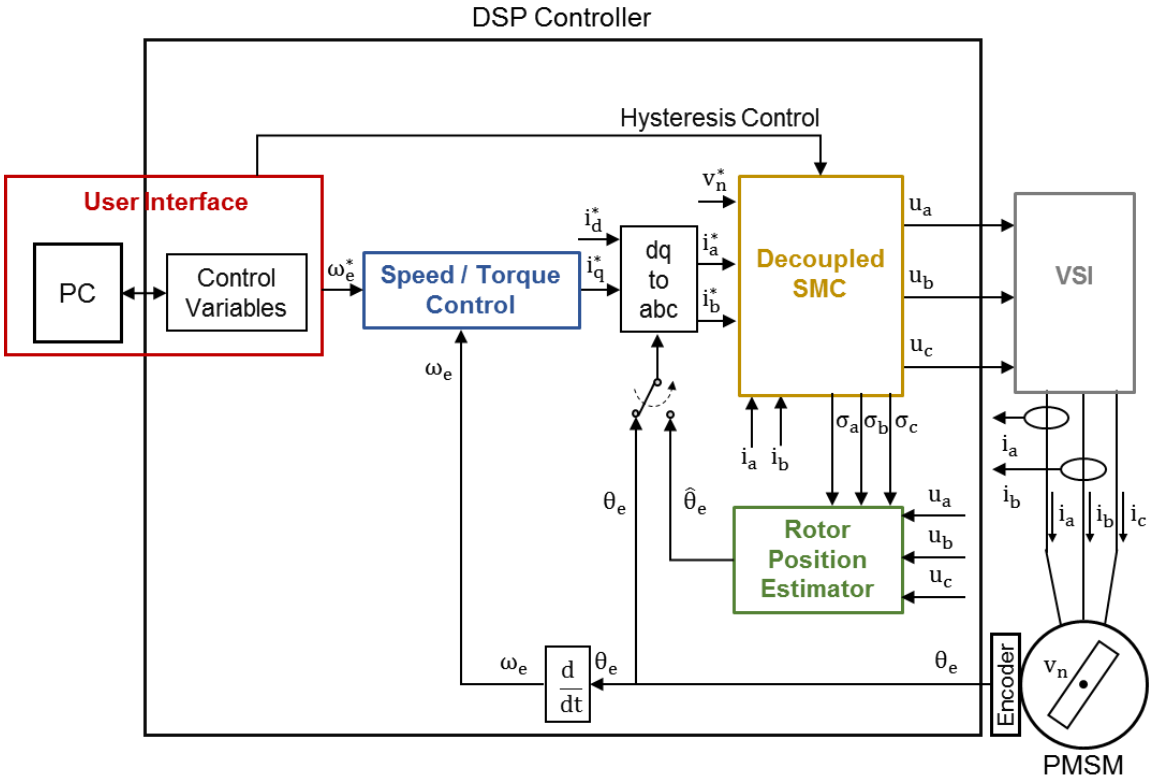


Figure 6-1: Control system block diagram

A personal computer (PC) with code composer is in communication with DSP to perform real time changings in the system. "Control Variables" block contains the variables that can be controlled during debugging; e.g. hysteresis control (fixed or variable), reference electrical speed etc. "Speed / Torque Control" block generates the reference current i_q^*. Normally, as we know, i_d^* is zero. "dq to abc" block converts the dq system to abc system and generates the reference currents i_a^* and i_b^*.

"Decoupled SMC" generates the control signals to operate the "VSI" and the same control signals are used in the estimator block. Also, it generates the sliding surfaces that are used in the estimator block. "Rotor Position Estimator" generates the estimated angle, $\hat{\theta}_e$. The angle is injected into the system in "dq to abc" block either measured by the encoder or the estimated one. The measured or estimated speed is injected into the "Speed / Torque Control" block (As we are not using the estimated speed in the feedback, it is not indicated in the block diagrams, neither in Figure (6-1) nor in Figure (6-2)).

A diagram that includes all the internal blocks of the whole system with more details is presented in the Figure (6-2).

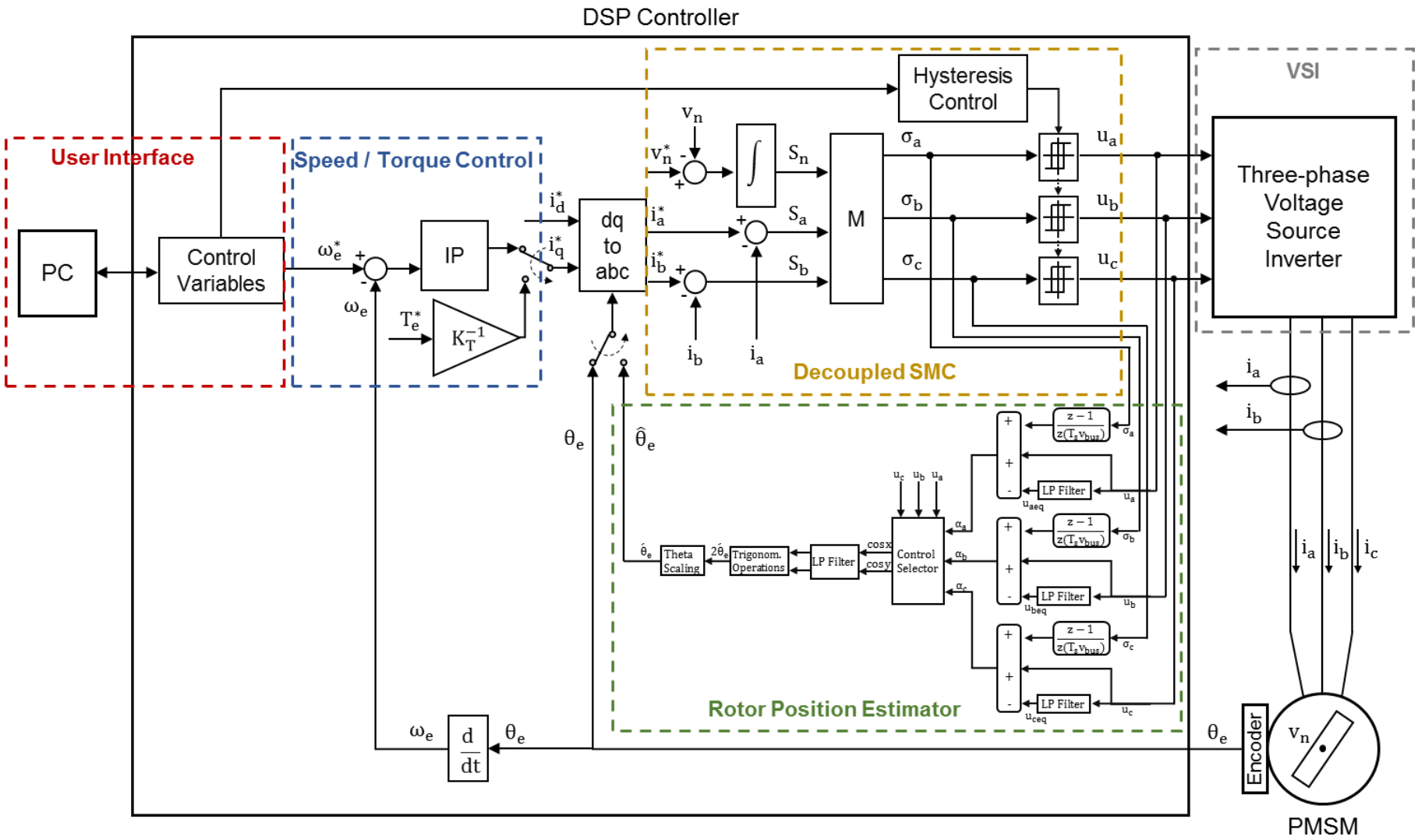


Figure 6-2: Complete control system block diagram

7. Implementation and experimental results

7.1. Fast and slow routines

As described before, in a real time system, there are always different kinds of tasks that can be performed faster or slower depending on the priorities assigned to them. The proposed algorithm plus SMC system are divided into two routines:

1. Fast execution routine
2. Slow execution routine

The SMC must be executed in the fast routine with highest priority so that there occur no delays, the control is performed properly and there are no disruptions while control part is being executed. On the other hand, there are some parts within this control that can be executed in the slow routine, for example; the speed controller which is implemented by an IP-Controller (IP-Controller is a bit different than the conventional PI-Controller) that can be executed in the slow routine because the mechanical dynamics of the motor are slower than the electrical one. Therefore, execution of the speed control in slow routine does not make a big difference.

Likewise, in the rotor position estimation algorithm, there are some parts that have to be executed in the fast routine and the other parts can be executed in the slow routine.

Hence, the estimation algorithm is divided in two subroutines that can be observed in the Figure (7-1).

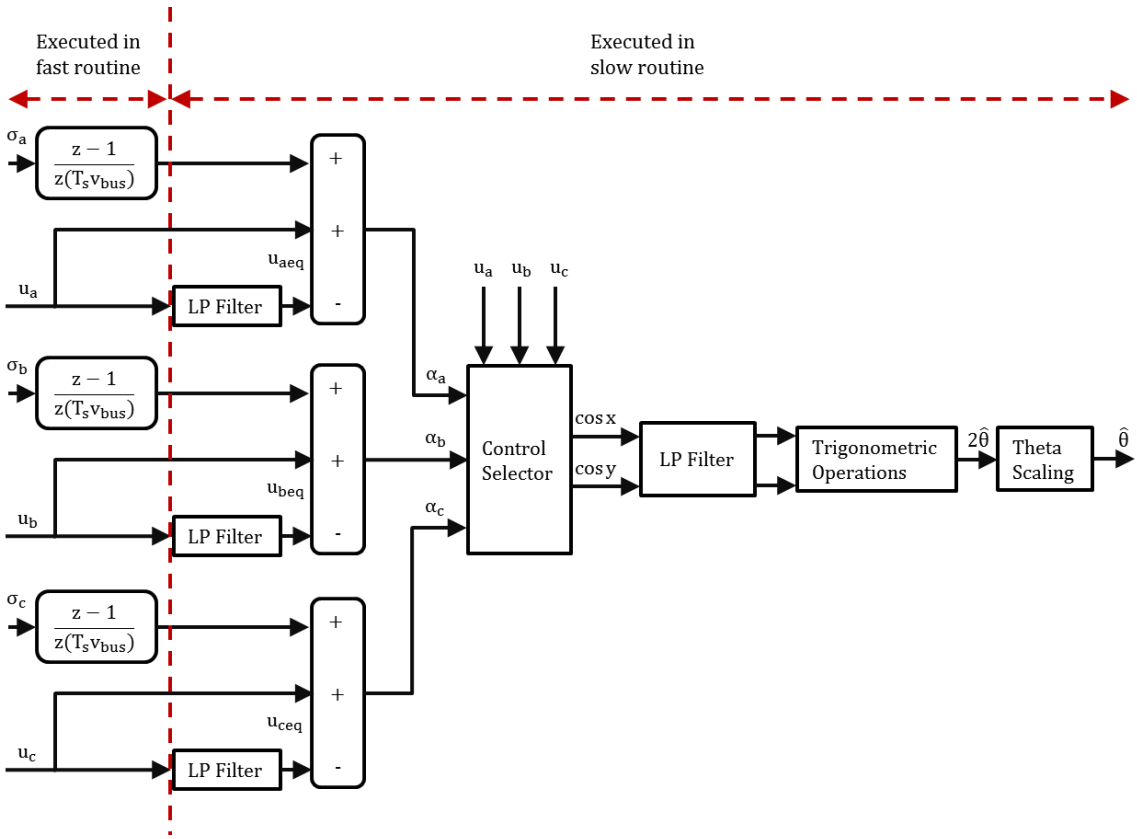


Figure 7-1: Rotor angle estimator algorithm division in subroutines

As it can be observed that the only part that is defined and has to be implemented in the fast execution routine is the derivative of the sigma signals, which are the decoupled sliding surfaces, designed in the equation (3.9), where in reality they are not fully decoupled due to the effect of the saliency that is always present in the system.

As sigma signals are extracted from the SMC block, as depicted in Figure (3-4), the information from these sigma signals must be extracted with the same sampling time as SMC. Thus with 6 µs it has been possible to extract the information good enough to perform the estimation in the real system. On the other hand, once the derivative of the sigma signals is performed, this information could have been further processed in the slow execution routine with 150 µs execution time.

7.2. Algorithm improvements during implementation in DSP

7.2.1. Encountered problems and their empirical solutions

7.2.1.1. Control signals synchronization

When the control was executed in the DSP system, the same problem described in Chapter 5 was encountered. By using the same solutions described in that Chapter, we have been able to desynchronize the control actions that made it possible to have rotor position information available most of the time to estimate the angle during experimentation.

7.2.1.2. Computation time

One main problem that was encountered during the implementation in the real system is the computation time of the DSP. As described before, in Chapter 5, topic 5.1, in Note III, two main factors affected the DSP computation time: big amount of c code to execute the SMC and the older version of DSP technology. Hence when tested for the initial desired subroutines execution timings, the DSP required more time to execute the complete system, in other words DSP requires more time to compute all the desired signals for the complete system. Therefore, after finding that problem, the subroutine execution time was adjusted empirically in the real system, in code generation from the MATLAB and once ensured that with the new timings the DSP system was able to compute all the signals properly without decreasing the efficiency of the SMC execution, the new timings were kept as, 6 µs for fast routine and 150 µs for slow routine and all the discrete time simulations were performed again.

7.2.1.3. Sigma signals slope measurement optimization

Since the beginning, the derivation of the switching surfaces was performed, and all the data generated by the derivation was saved all the time. Starting with continuous time simulations and then in the discrete time simulations as well. Until we arrive to the DSP system and found the computation time problem.

At that point after performing an analysis on the sigma signals, we realized that the change in the slopes that occurs in the sliding surfaces, as shown in the Figure (3-3), happens when one or more than one or any of the control signals (ua, ub, uc) changes. Therefore, there was no need to continuously perform the derivation operation and save all the information all the time. Hence, just by detecting the change in the control signals and only performing the derivation in the moment when any of the control signals commutates and then save the new information only and keep it until the next control commutation happens

is enough. Furthermore, we obtain the same information from the sigma signals as before but at the same time the computation time for the DSP to execute this part of the estimator is reduced.

7.2.1.4. Theta estimation error divergence

Second main problem that was encountered during the implementation of the algorithm in the DSP system was the divergence of the theta error, as it appeared initially, that didn't happen in the simulations but, it surely did. Since the simulations were only performed for 1 s time it was never possible to observe the theta divergence problem as for longer simulations it requires a huge amount of time and very power full operating system to run the simulations in the desired realistic conditions. Where this problem was found in the real system after running it for more than 1 or 2 minutes, which of course is a big time to perform discrete time simulations in real conditions.

Knowing that there is an integrator in the last part of the theta scaling block, as shown in the Figure (3-7), this integrator was continuously accumulating the error.

To sort out this issue, according to the research work presented in [32], when an angle position is the real motor angle position, the generated direct current, i_d goes to zero. Thus, by applying this principle and adjusting the gain a loop across the integrator is being closed making it track the real angle in steady state, where in transients there occur some small lag/lead in the real angle tracking but once the system is in steady state estimation tracks the real angle and does not diverge.

Note V: *This improvement is not added into this work as it needs to be studied thoroughly with mathematical proofs and simulations of how this part of the system works exactly. However as learned from the paper and tested empirically in our system, it has been possible to stop the divergence of the error and theta estimation converges to the real angle always. Results obtained with the real system are presented later in this Chapter.*

7.2.1.5. 1st order digital low-pass filter implementation

A DSP can execute 1st and 2nd order digital filters very fast in a very small computation time. However, due to the computation time problem that we already had with our DSP system, it was not possible to implement even a 2nd order digital filter.

Graph in the Figure (7-2) shows the response of the 1st order digital filter.

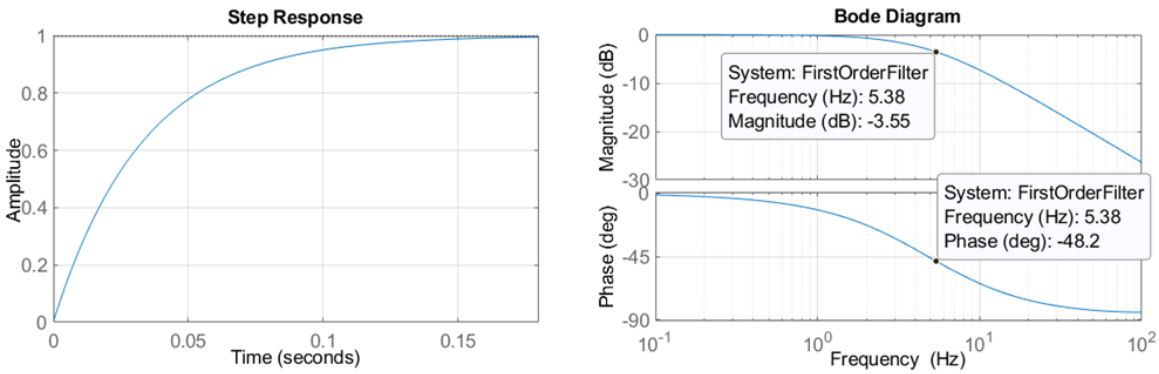


Figure 7-2: 1st-Order digital filter response

Hence the order of the filter had to be reduced and the coefficients must be redesigned to execute it in the DSP system to perform the angle estimation. The transfer function of the new filter can be expressed as following:

$$\mathrm{TF}_{1\mathrm{st\_order\_digital\_filter}} = \frac{0.0045\mathrm{z}}{\mathrm{z} - 0.9955}$$

As it can be observed, the new -3dB cut-off frequency of the filter is around 5 Hz, where in the worst case scenario, ±30 rad/s, the signal that appear is on 9.6 Hz frequency as discussed before in the Table 6, which is almost at the twice of the cut-off frequency of the new filter.

Therefore, to filter maximum noise and at the same time to be able to estimate the angle properly, these coefficients of the filter have been chosen with an empirical; hit and trial method.

And it can also be observed that in the worst case scenario, the filter is adding a maximum phase shift of around 60 degrees to the estimation. This might be another problem in the estimated angle. But thanks to the solution described in previous topic 7.2.1.4, “Theta estimation error divergence”, with its implementation the phase error in the estimation is not a problem anymore, system automatically corrects the phase and within a finite time, estimation converges to the real rotor position angle value. Some results obtained from the real system tests can be seen in the topic 6.5, “Experimental results”, where it can be easily observed how the system automatically corrects the phase shift in the estimation.

7.3. Hardware platform

Following snapshot Figure (7-3) describes the complete control system board.

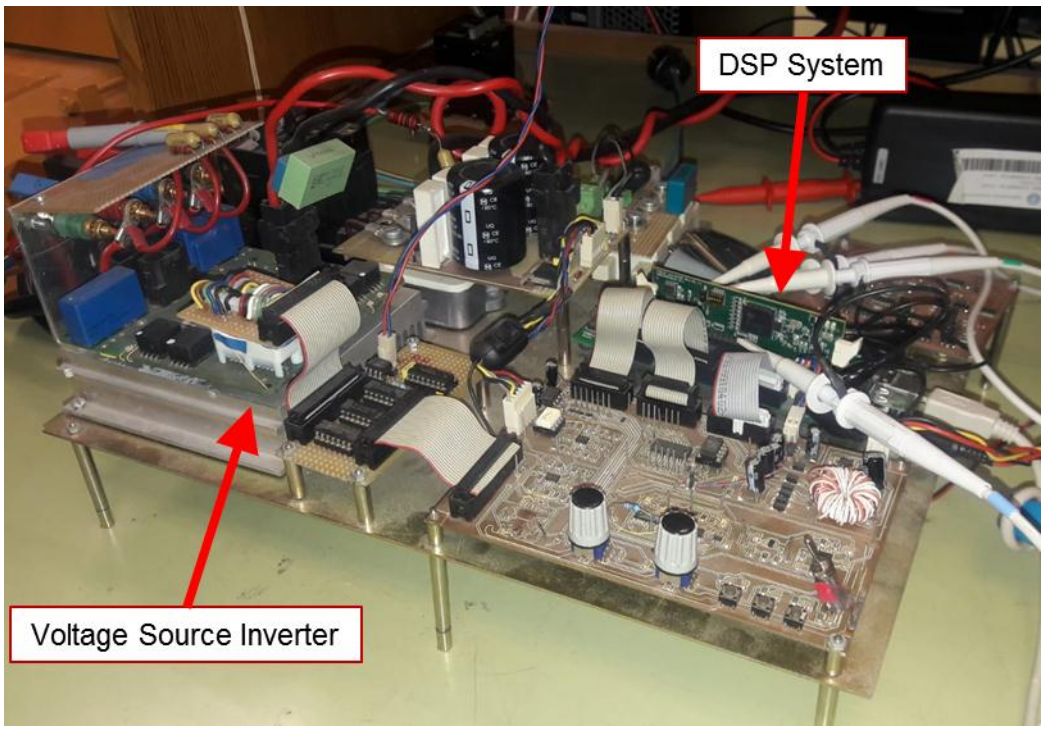


Figure 7-3: Data acquisition board

The DSP controller model is TMS320F28335: Delfino™ 32-bit MCU with 150 MIPS, FPU, 512 KB Flash, EMIF, 12b ADC and for VSI using a 3 phase IGBT to control the motor and

the Oscilloscope model is Yokogawa DLM2024 mixed signal oscilloscope pointed in the Figure (6-5).

The PMSM drive used can be seen in the following snapshot Figure (7-4):

Figure 7-4: Permanent magnet synchronous motor drive

A snapshot of complete hardware including; machine, control system board, oscilloscope and power supplies are shown in the Figure (7-5):

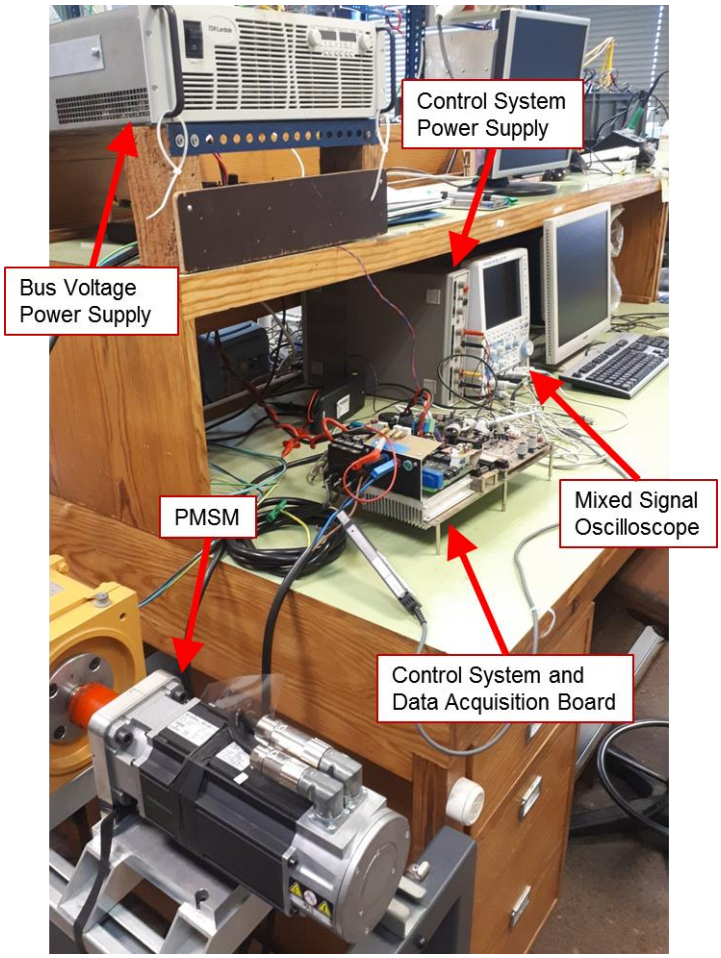


Figure 7-5: Complete hardware setup

7.4. Extraction of the code from SIMULINK MATLAB

Following are the steps that need to be followed to generate the code from the MATLAB, of a Simulink model and run it in a DSP system.

1. First, in the model configuration parameters of the SIMULINK model, user has to configure all the parameters according to the DSP model that user wants to use and save the settings.
2. Once the modelling of the system and control in the SIMULINK is ready the next step is to generate the code and export it to hardware.
    a. If there are no errors and everything goes well, MATLAB will generate the c code for the hardware.
3. Once the build is finish user can open Code Composer Studio and locate the built project by the MATLAB.
    a. An alternative way is to open the diagnostic viewer in the MATLAB and locate the link of Code Composer Studio and open the built project directly.
4. In Code Composer Studio in the project structure user can view and modify all the files generated by the MATLAB including the c code if necessary.
5. Then user must enter in the project folder properties and configure all the parameters according to the DSP model that user wants to use and save the project settings.
6. Next step is to build the project by clicking on the build button.
    a. In this step, Code Composer studio readjusts some parameters in the files according to the new project settings.
7. Then next step is to generate a debug configuration file and use the files generated by the project into the debug configuration tabs and verify the processor is activated and save it.
8. Finally, user can load the code into the hardware by using debug/debug and run button.

If all the mentioned steps are followed and process is done correctly there will be no errors and system will start running in the debug mode.

7.5. Experimental results

Experimental results are performed in the same manner as the results shown before in continuous time and in discrete time simulations. All tests are performed separately and instead of load impact a torque control test has been performed in both, open loop and sensorless operation. Where green graph represents the real angle obtained from encoder and blue graph represents the estimation with red graph that represents the estimation error in radians. The results obtained from the real system can be visualized in the following sections.

In the experimentation, θ_e and $\hat{\theta}_e$ are plotted in rad/s between 0 to 2π and they correspond to the ground / 0 V and 2.5 V respectively, with a voltage range of 1 V/div.

Therefore, the relation between the voltage and plotted θ_e is the following:

$$\theta_e = \frac{\mathrm{Vmes} * 2\pi}{\mathrm{Vmax}} \ \mathrm{rad/s}$$

where; Vmes is the voltage measured on the oscilloscope and Vmax is the maximum voltage of the range on which the signal is being plotted, that in this case is 2.5 V. The same expression is also valid for the $\hat{\theta}_e$.

To convert the angle from rad/s into degree, the conventional conversion relation can be used, which is;

$$\theta_e\ (\text{degree}) = \frac{180}{\pi}\ \theta_e\ (\text{rad/s})$$

The error in the following experimental plots is just the subtraction of the real and estimated theta, thus the same expressions as above can be used to calculate the error in either rad/s or degrees. However, the error values that have been calculated during the experimentation, from the voltage measure error lies between 2 degrees and 15 degrees in steady state and in transients it reaches up to a peak of 36 degrees due to the phase added by the filter. However, after the transients the error converges to its steady state value within finite time which is worst case has been measured between 2 s and 2.5 s. Thanks to the dynamics that are implemented in Topic 7.2.1.4, the estimation error always converges and tries to reach zero.

7.5.1. Estimator performance in open loop operation

7.5.1.1. System start-up test

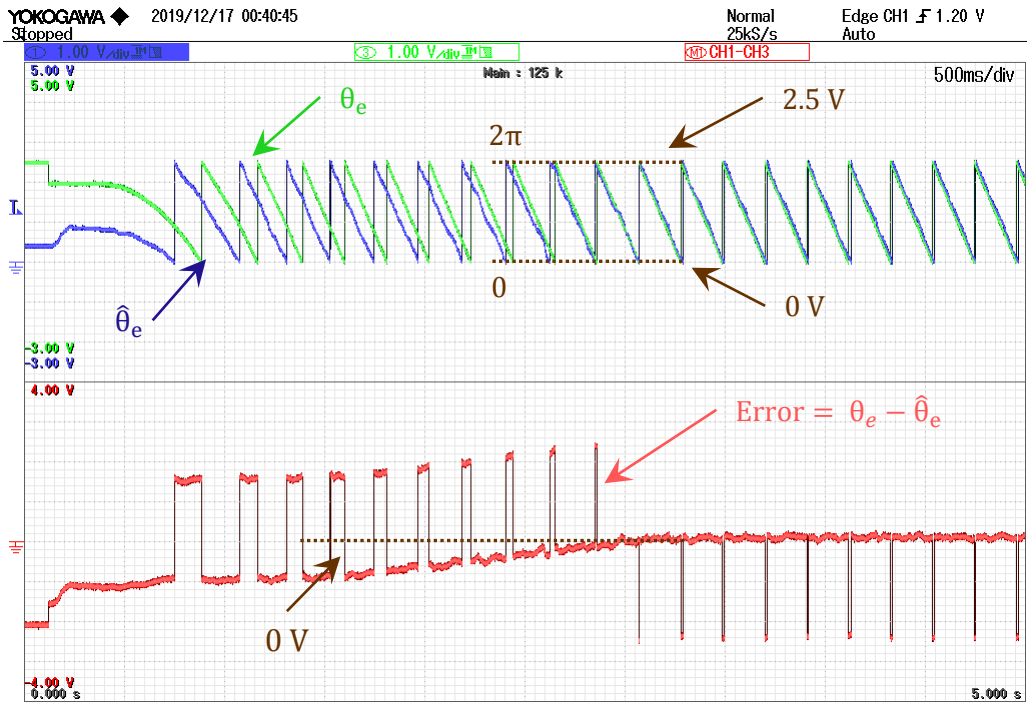


Figure 7-6: System start-up and angle error convergence in open loop

Figure (7-6); At start-up, angle estimation (in blue) converges to real angle (in green) within 2.5 s. Angle error (in red) in steady state is around 0.1 V or ~14.4 degrees.

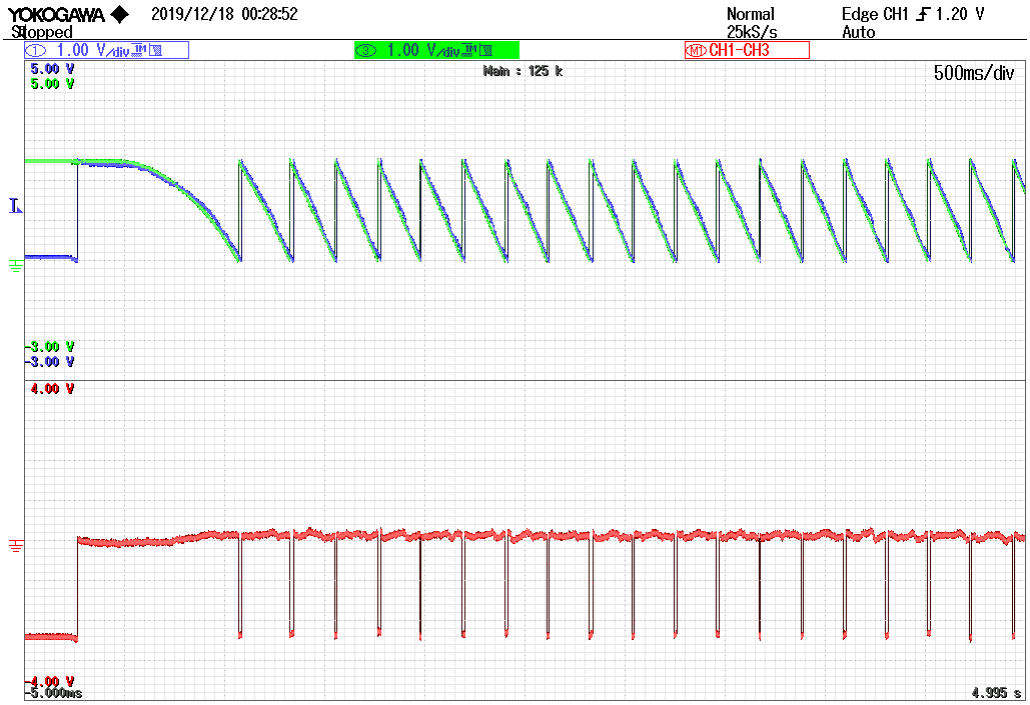


Figure 7-7: System re-run with different initial condition in open loop

Figure (7-7); start with any initial condition after the system start-up, angle estimation always tracks the real angle. After the transient angle error stays constant all the time.

7.5.1.2. Speed reversal test

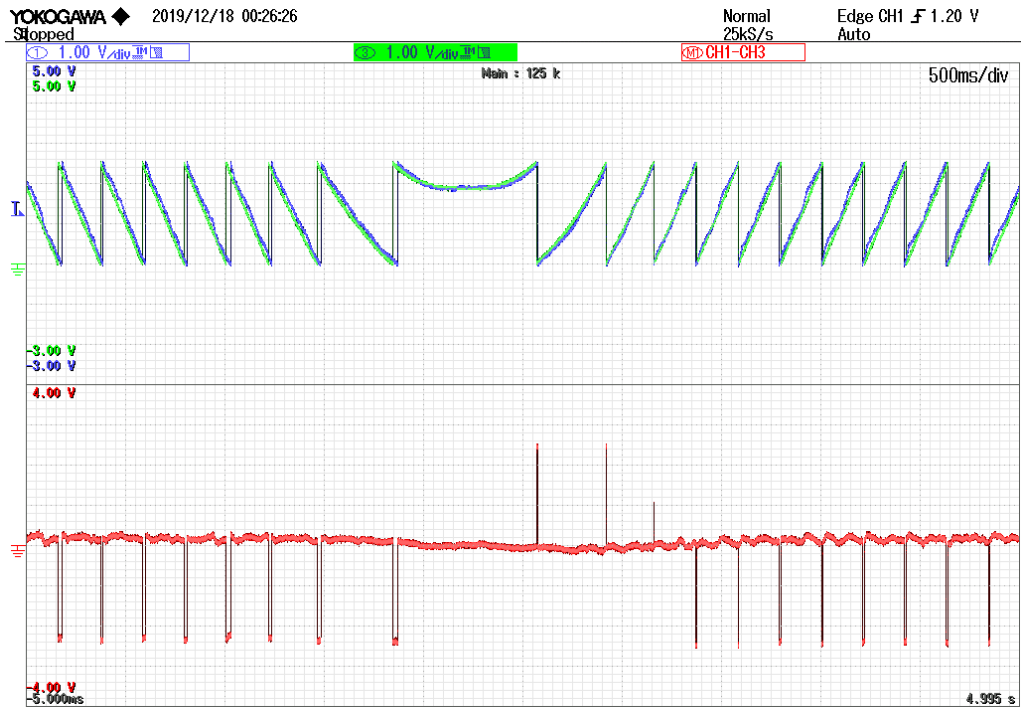


Figure 7-8: Angle estimation for speed reversal in open loop

Figure (7-8); on speed reversal, estimation (blue) tracks the real angle (green), and the angle error stays constant before and after the transient around 0.1 V or ~14.4 degrees.

7.5.1.3. Zero speed test

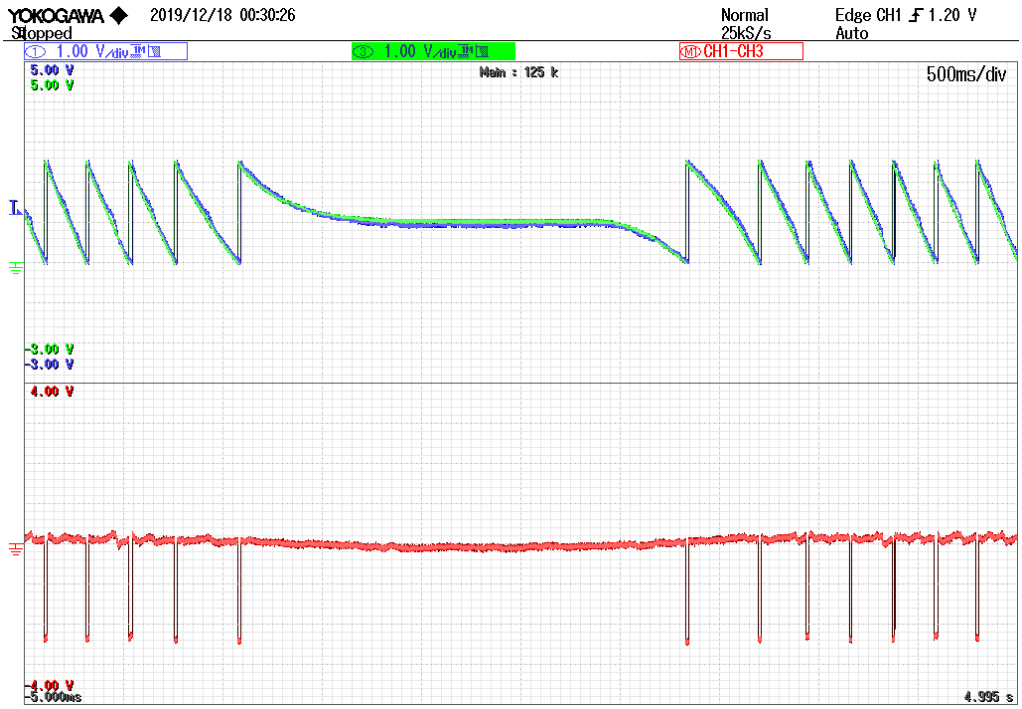


Figure 7-9: Angle estimation for -30 to 0 to -30 rad/s speed test in open loop

Figure (7-9); proved the angle estimation at 0 speeds. Start with -30 rad/s then speed is set to 0 rad/s, estimation tracks the real one with zero error then back to -30 rad/s.

7.5.1.4. Torque control test

Figures (7-10) and (7-11) shows the estimator response during the torque test.

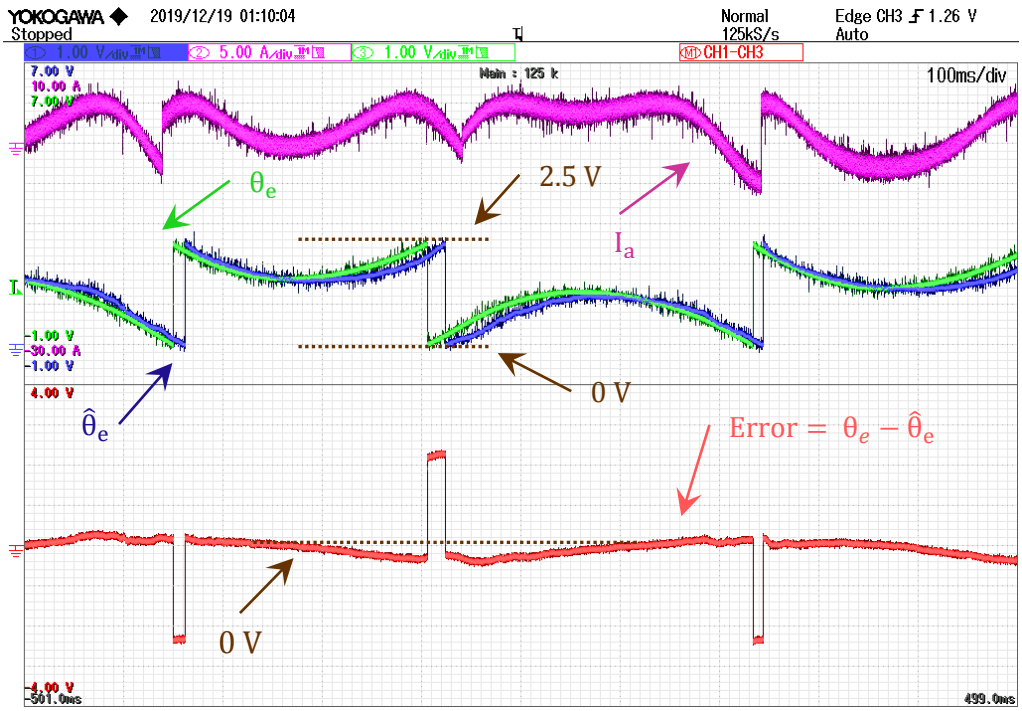


Figure 7-10: Angle estimation for torque control test in open loop

Assuming known the K_t motor constant (relation between i_q and electromagnetic torque), a direct torque control can be implemented controlling i_q. Therefore, a square wave torque reference is applied directly to i_q reference through gain K_t. With i_q reference, system generates very fast transients in clockwise and counter clockwise directions.

Thus, those fast transients are worst case scenarios to test the estimator. On the other hand, this test is very stressful for the motor drive itself, so it not recommended to continuously performing it for longer times.

Figure (7-10); a phase current of the motor (in magenta), real angle (green) and estimation (in blue) can be observed with angle error (in red) which is around 0.25 V or ~36 degrees in worst case. It can be noticed that how accurately angle estimation is tracking the real angle of the motor measured by the encoder. Considering that there is a phase delay in the estimation, system continuously estimates the angle in these worst conditions.

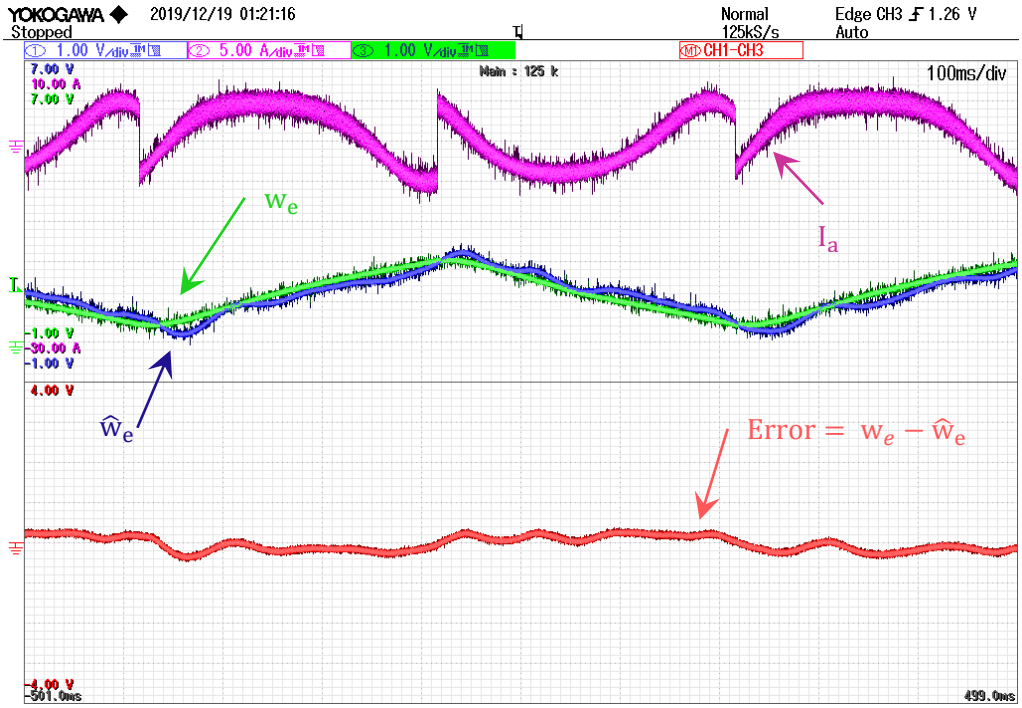


Figure 7-11: Speed estimation for torque control test in open loop

Figure (7-11); a phase current of the motor (in magenta), real speed (green) and estimated speed (in blue) can be observed with speed error (in red). As expected the speed estimation is not good enough, even though in these worst conditions the speed estimation still tracks the real speed.

7.5.1.5. Speed estimation test

The graphs in the Figure (7-12) show the estimated speed from real system.

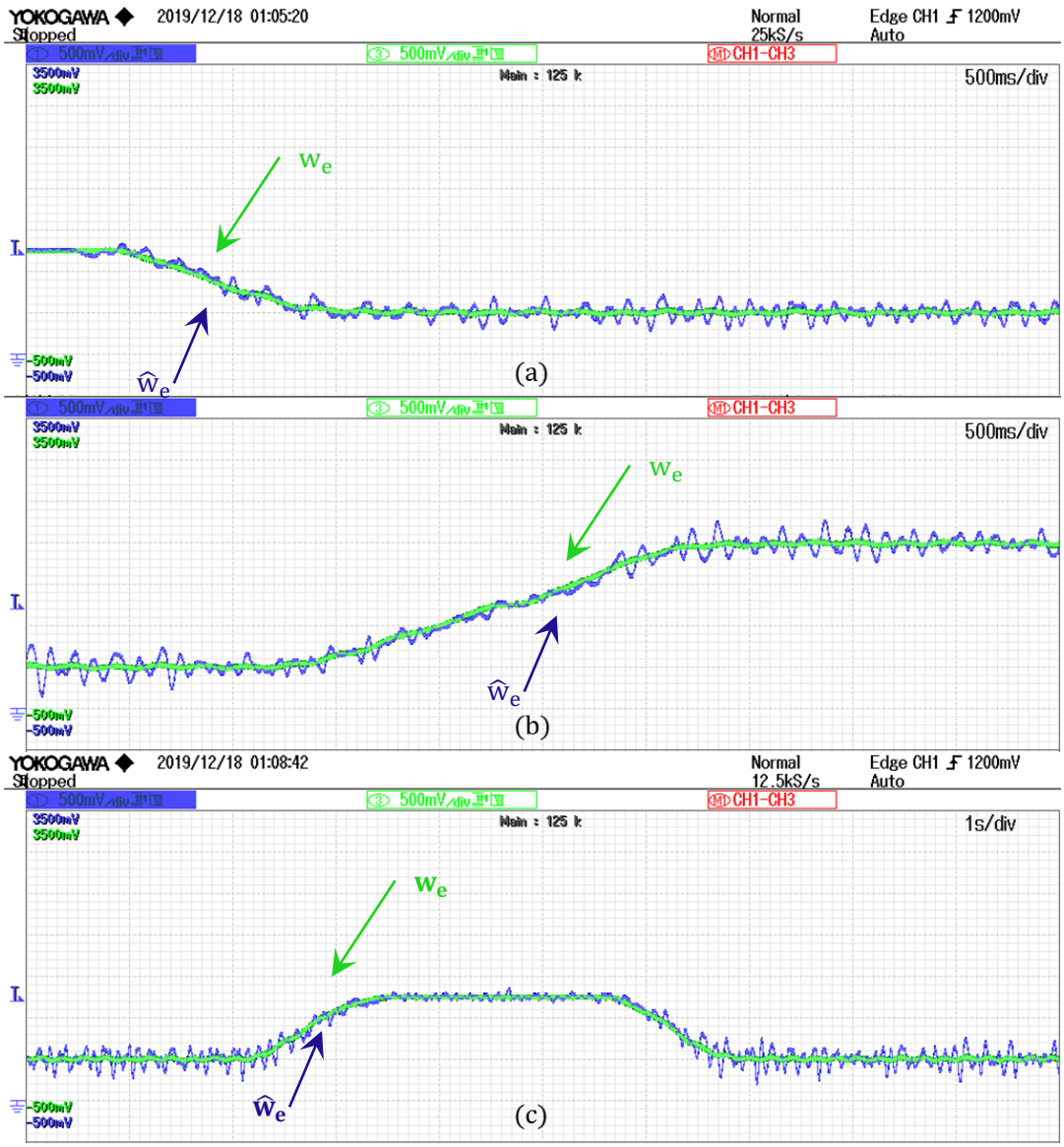


Figure 7-12: General speed estimation response in open loop

where; (a): represents the start-up speed transient, (b): speed reversal transient and (c) -30 to 0 to -30 rad/s speed transient.

As suspected, it can be observed that speed estimation is not good enough. In fact compare it with Figure (4-7) in open loop, there is similar noise in the system but worse in terms of ripples.

7.5.2. Estimator performance in sensorless operation

Before switching to the sensorless operation, the loop has been closed during the steady state function of the system in order to see the behaviour of the estimator and angle error.

Following section represents the results obtained with this operation.

7.5.2.1. Sensorless operation activation in steady state

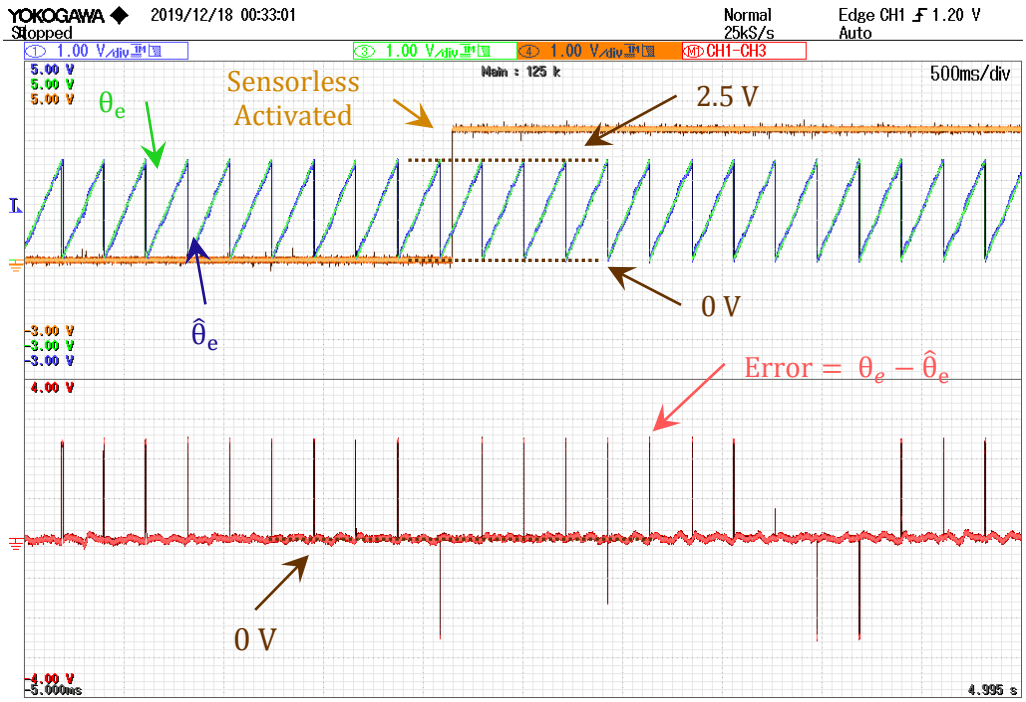


Figure 7-13: Angle estimation on sensorless activation in steady-state

Figure (7-13 and 7-14); Orange signals represent the activation of the sensorless mode in the middle of the graphs. It can be verified that before and after the activation, there is no difference in the angle and speed estimation and system works in sensorless mode with the same error, thus no error divergence happened.

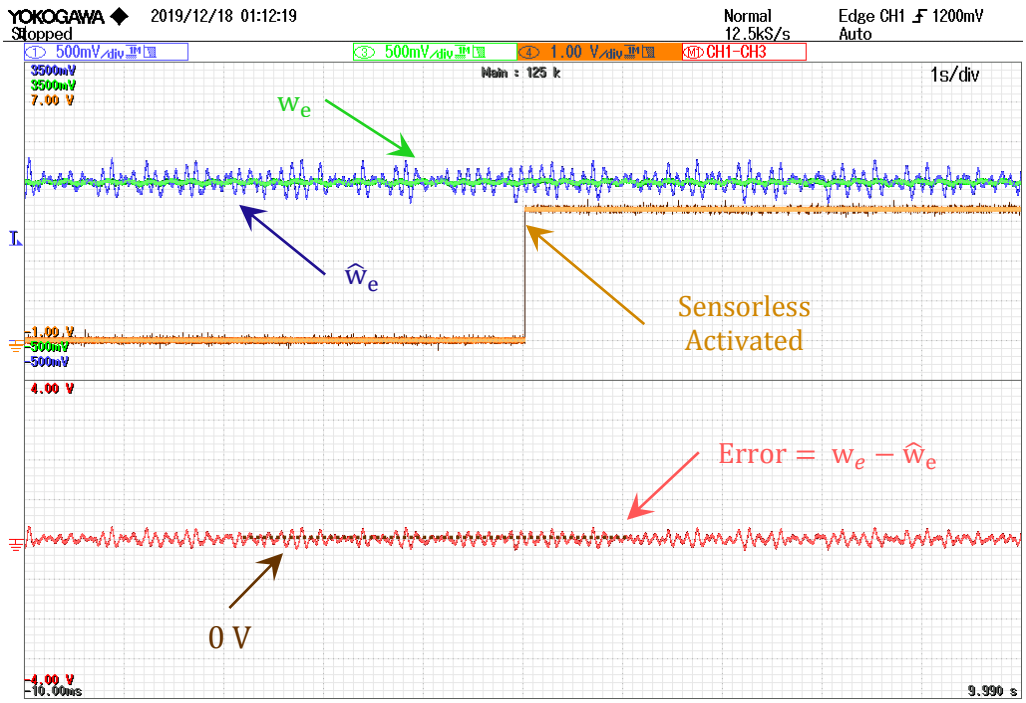


Figure 7-14: Speed estimation on sensorless activation in steady-state

7.5.2.2. System start-up test

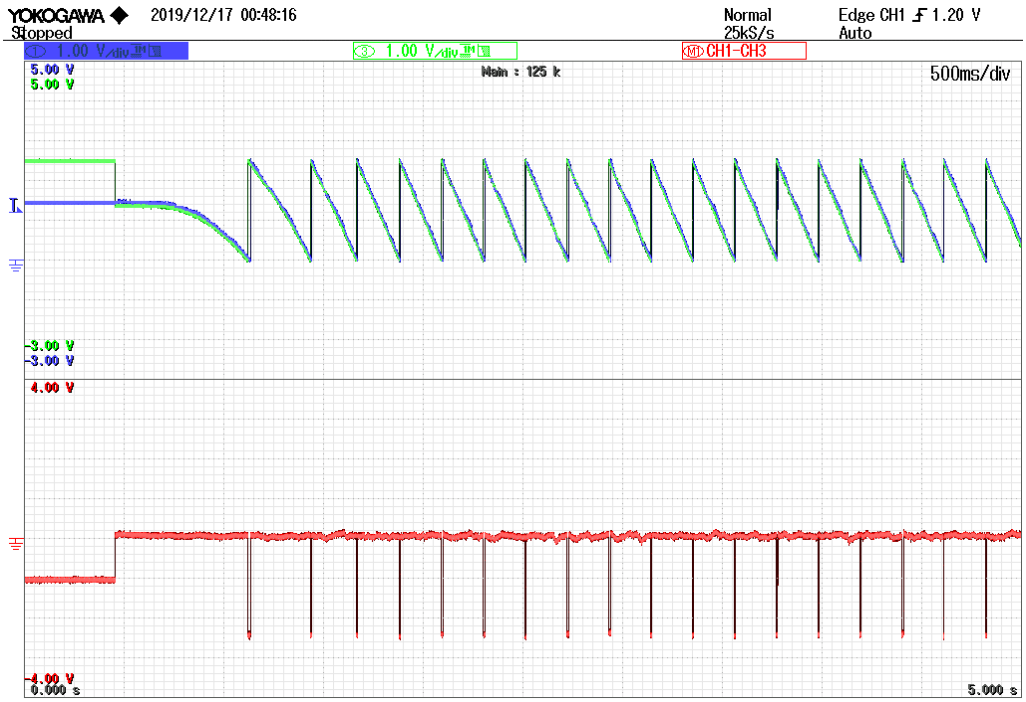


Figure 7-15: System start-up and angle error convergence in sensorless

Figure (7-15); At start-up, angle estimation (in blue) tracks the real angle (in green) without any delay. Angle error (in red) is constant around 0.1 V or ~14.4 degrees.

Figure 7-16: System re-run with different initial condition in sensorless

Figure (7-16); Start with any initial condition as system is already in sensorless mode, angle estimation always tracks the real angle as in Figure (7-7) with an angle error around 0.15 V or ~21.6 degrees and it does not diverge.

7.5.2.3. Speed reversal test

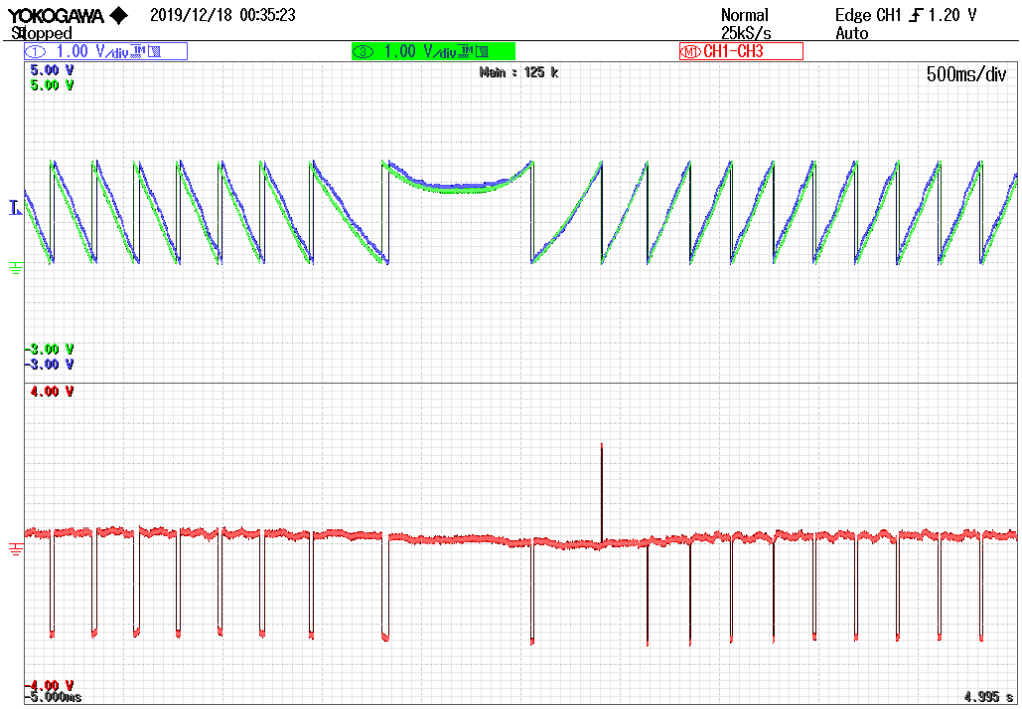


Figure 7-17: Angle estimation for speed reversal in sensorless

Figure (7-17); On speed reversal, estimation (blue) tracks the real angle (green). Angle error is less than 0.15 V or ~20.1 degrees, a bit more than the open loop test Figure (7-8).

7.5.2.4. Zero speed test

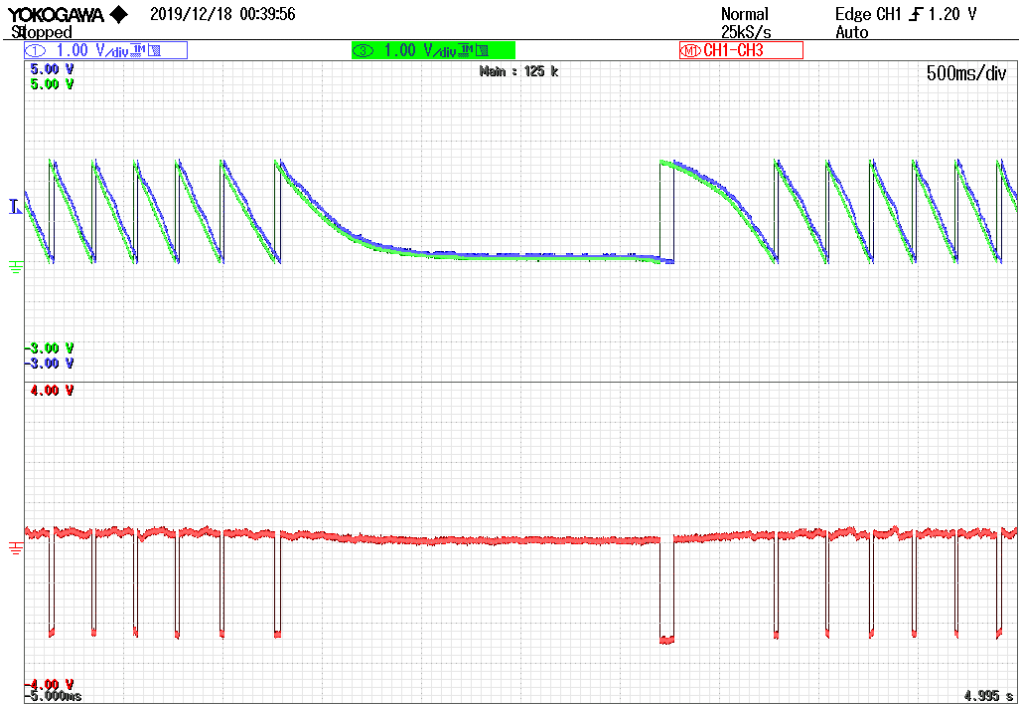


Figure 7-18: Angle estimation for -30 -0- -30 rad/s in sensorless

Figure (7-18); At 0 speed in sensorless estimation tracks the real angle that proves the biggest objective of this estimation technique. Phase error is a bit more, but it is constant.

7.5.2.5. Torque control test

Figure (7-19) and Figure (7-20) shows the torque transient test for angle estimation and speed estimation respectively.

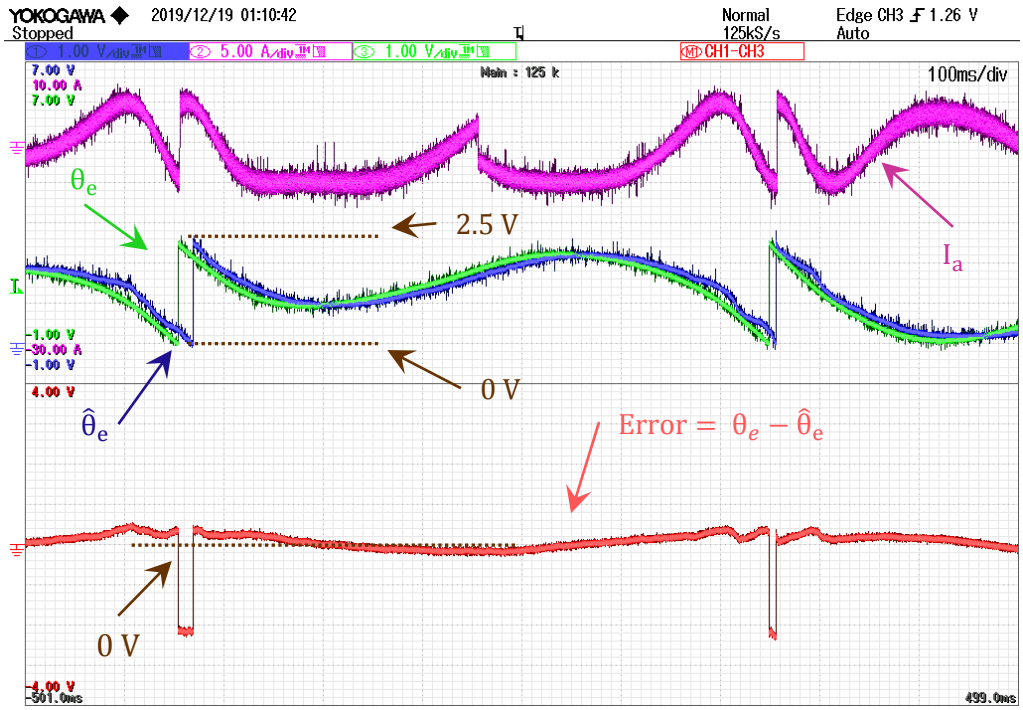


Figure 7-19: Estimation angle response for torque control test in sensorless

Figure (7-19); A phase current (in magenta) as before and comparing it with open loop result presented in Figure (7-10), similar angle error around 0.25 V or ~36 degrees in worst case can be observed.

Figure 7-20: Speed estimation response for torque control test in sensorless

Figure (7-20); Shows the similar results as presented it in open loop Figure (7-11).

7.5.2.6. Speed estimation test

The graphs in Figure (7-21) showss the estimated speed in sensorless operation from real system.

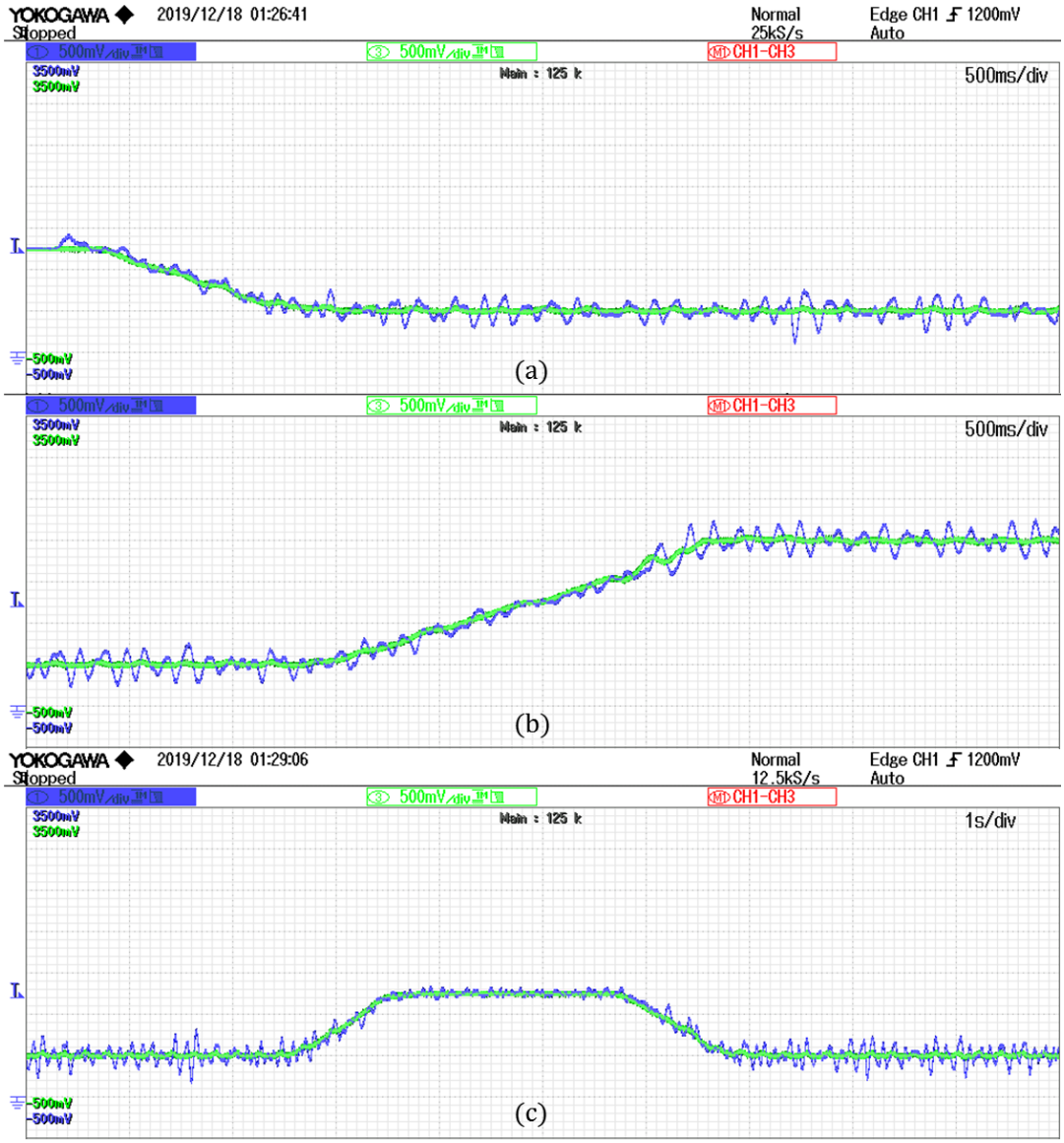


Figure 7-21: General speed estimation response in sensorless

As depicted in discrete time simulation speed estimation in open loop result presented in Figure (5-8) and in sensorless operation in Figure (5-14), the noise in the estimation increases in sensorless operation. Therefore, the same results can be observed in the estimated speed in the real system. Comparing Figure (7-12) and Figure (7-21) it can be visually verified that the noise in the sensorless operation is greater than the open loop operation.

7.6. Comparison

Comparing the overall results in both open loop and sensorless operations from experimentation, it is verified that the data provided by the angle estimation is very

promising. Knowing that the angle estimation error is bigger than expected in the real system, the system continues to converge the estimated angle to the real one all the time. These results can be improved by changing the existing hardware system. Hence with the presented results, the implementation of the proposed estimator in a real system is verified.

8. Conclusions and future developments

8.1. Conclusions

In this presented work, thanks to the implementation of a decoupled SMC which is designed by neglecting the effect of the saliency we have been able to detect the effect of the saliency in the decoupled surfaces. The surfaces are not piecewise linear anymore due to the saliency of the motor and the idea of extracting the position signals information from those sigma signals was born which then led us to prove theoretically, that it is possible to extract the information from the system by implementing a control selector as depicted before in Chapter 3 and then, from there, rotor angle position can be estimated.

One major part of the main objective of this work, as I have studied, designed and implemented the proposed rotor angle estimator in continuous time in both open loop and sensorless operations. By the help of simulations, I have obtained very promising results, presented in Chapter 4 that proves the angle estimation and the feasibility to further implement it in discrete time.

Furthermore, the second main objective, I have discretized the system according to the parameters and changings presented in Chapter 5 considering the maximum limits of the available DSP system and not leaving behind the worst case scenario for the rotor angle estimator. On the other hand, considering the minimum and maximum speed limits according to the existing estimator systems presented in literature that work for high speeds, to cover the low speed areas and even a stopped motor. Thus, with the help of simulations in discrete time, I have proven this system can be executed in the real time and angle estimation can be performed with the available hardware. Then by comparing the continuous time and discrete time results I have verified the increase in noise in the signals due to fact of discretization.

Finally, by dividing the designed estimator into two parts, for fast and slow execution routines and by improving some parts empirically I have been able to execute it in the DSP system for both; open loop operation and sensorless operation. That proves the viability of the estimator without using a very high performance system and obtained the angle estimation results from the system on oscilloscope which are presented in Chapter 7. That verifies the system implementation in a real system and sensorless operation that has been performed not just for the worst case speed scenario but also for a stopped motor, at 0 rad/s speed. This test finally proves the main objective of the estimator; that is to estimate the rotor angle position all the time even for a stopped motor without injecting anything else into the system.

Hence by comparing the results obtained in Chapter 5 and Chapter 7, it has been verified that the real system does work as designed for discrete time simulations if considered all the real time parameter during discretization. The results obtained from the real system are more than 95% like the ones obtained in the discrete time simulations.

Moreover, the results obtained in the real system for speed estimation are also like the ones obtained in discrete time simulations. This of course can be improved in the future study of this work.

Furthermore, as described in Chapter 5 and Chapter 7, due to the computation time problem some parameters of the predesigned estimator have been changed. Therefore, it has decreased the performance of the estimator. These parameters could be improved, and

performance of the estimator could be much better. More information regarding this topic is provided in the next section “Future developments”.

Being said all of that; it would not be too much to say that this work is a significant contribution in the angle estimators / observers part of this field of study. It has been verified during the research that; there does not exist any work which is presenting this kind of estimator, even though there are works which are using saliency effect to measure the angle. That means this is the first of its kind angle estimator for PMSM that has been designed and implemented in a real system proving its viability without decreasing the system efficiency and with no need of injection of any kind of voltage or current signals into the motor. Most of all the rotor position is estimated all the time even at zero motor speed. Hence the proposed estimator is a huge contribution in the literature of the race in which estimating the rotor angle position of a PMSM without an encoder is a norm.

8.2. Future developments

First, the most important thing that needs to be changed is the DSP hardware system. By doing so, one of the major problems with the existing real system would be solved and this will result in more available computation time to improve the presented estimator by playing with more ideas for the improvements.

Once there is more computation time available, the parameters of the estimation algorithm that has been lower down for the sake of the estimator to just work in the existing system, those parameters could again be upgraded, and system must be tested in the first desired designed conditions e.g. with computation times; 5 µs for fast execution routine and 125 µs for slow execution routine of the system. Not only that, the 2^{nd} order digital filter, shown in Chapter 5, topic 5.5, which has been changed with a 1^{st} order digital filter, shown in Chapter 7, topic 7.2.1.5, must be changed again with 2^{nd} order digital filter and results must be tested in terms of noise in position signals estimation.

On the other hand, as it has been verified in the Table 6, that the position signals moves across the frequency spectrum depending on the electrical speed of the motor with a maximum bandwidth of 2 Hz as it is tested, thus it is very advisable and recommended that instead of using a digital low-pass filter to filter the position signals, design and implement a digital band-pass filter (BPF) with higher order and adoptable parameters depending on the motor speed with a 2 Hz bandwidth. By implementing that, theoretically and of course practically, all the noise present in the position signals would be filtered and almost perfect sine and cosine signals would be extracted from the system which then will result in almost ideal angle estimation or at least as same or clean as the angle measured by an encoder itself.

Note VI: *A test with a higher order digital BPF have been made but due to the shortage of the time and the problem of the computation time of the DSP, those results were not captured, and they are not added into this work. Nevertheless, as analysed by both: author of this work and his advisors, by using a BPF with small bandwidth as described above, a perfect angle estimation has been already verified on the oscilloscope.*

Therefore, one of the major improvements that could be done with the presented estimator algorithm is to replace the low-pass filter with the described or similar band-pass filter.

Also, the speed estimation must be improved, as described before, the source of all the information that is being used to estimate the angle and speed is coming from position signals. It can be easily predicted that once the position signals are cleaner than the ones

presented in this work, automatically, as angle results in better estimation, the speed estimation will also start to improve. Furthermore, to enhance the speed estimation even more, another low-pass filter could be designed and implemented, and the results would be tested.

Other than that, another area to work in this algorithm is to obtain the initial rotor position when starting up the system with powering on. Because as demonstrated before once the system is in working condition and it already has initial condition, if the control system has power, it will not lose the current position of the rotor. But as demonstrated before in Chapter 2, there are works whose author's only intention is to get the initial rotor position on the system powering up. It would be an interesting topic study as presented way of rotor angle estimation is different than any work presented in the literature except for the fact that it also uses the information coming from the saliency effect.

In the last part of this estimator, in scaling theta bock, the new part that has been added to converge the estimated angle to the real one as described before in the topic 7.2.1.4, needs to be studied thoroughly with full mathematical proofs of the equations and simulations results and then this part in the estimator must be readjusted according to the theoretical results. Following the same procedure as followed by the author of this work, starting with the continuous time simulation then simulations in discrete time and comparing the results obtained in both cases and comparing them with the results obtained in this work to exactly conclude the difference in the estimation and error.

As it has been stated several times in this work, the focus of the design of this algorithm is to work for low and zero speeds. That brings up another interesting area of study that is to use the presented algorithm for the complete range of a motor speed. It might be possible that with some modifications in the presented estimator (e.g. by reducing the sampling times of the fast and slow subroutines or in other words increasing the sampling frequency in new DSP hardware, because the existing DSP system cannot faster than that), it would be possible to design a fully stand-still estimation system that would be capable of estimating the angle for the full speed range with no requirement of encoder at all.

Therefore, there are several parts of the presented algorithm that can still be improved, which in the first place was not possible due to the problem of the computation time in the real system. On the other hand, as stated, there are several areas for study related with this estimator in which some new parts can be added to the existing solution or some focused parts of the algorithm can be redesigned to implement it as a stand-still rotor angle position estimator.

Bibliography

[1] Paul C. Krause; Oleg Wasynczuk; Scott D. Sudhoff. ANALYSIS OF ELECTRIC MACHINERY AND DRIVE SYSTEMS, SECOND EDITIO OR THIRD EDITION. https://www.academia.edu/35755650/ANALYSIS_OF_ELECTRIC_MACHINERY_AND_DRIVE_SYSTEMS

[2] Vladislav M. Bida ; Dmitry V. Samokhvalov ; Fuad Sh. Al-Mahturi. "PMSM vector control techniques — A survey" 2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus).

[3] Repecho del Corral, Víctor, "Técnicas de inyección para la estimación de ángulo en un PMSM bajo control deslizante". M.S. thesis, Departament d'Enginyeria Electrònica. Universitat Politècnica de Catalunya, 2011.

[4] Bian Yuanjun ; Guo Xinhua ; Song Xiaofeng ; Wu Yanfeng ; Chen Yin. "Initial rotor position estimation of PMSM based on high frequency signal injection". 2014 IEEE Conference and Expo Transportation Electrification Asia-Pacific (ITEC Asia-Pacific).

[5] Yuanlin Wang, Xiaocan Wang, Wei Xie and Manfeng Dou. "Full-Speed Range Encoderless Control for Salient-Pole PMSM with a Novel Full-Order SMO", article, 2018. https://www.researchgate.net/publication/327634712_Full-Speed_Range_Encoderless_Control_for_Salient-Pole_PMSM_with_a_Novel_Full-Order_SMO.

[6] A. Ravikumar Setty ; PurnaPrajna R Mangsuli ; Kishore Chatterjee, 2017 International Conference on Power and Embedded Drive Control, State of the art and future trends in IRPE of PMSM using high frequency signal injection techniques.

[7] Wu, R.; Slemon, G.R., A permanent magnet motor drive without a shaft sensor, IEEE Trans. Ind. Appl. 1991, 27, 1005–1011.

[8] Sepe, R.B.; Lang, J.H. Real-Time Observer-Based (Adaptive) Control of a Permanent-Magnet Synchronous Motor without Mechanical Sensors. IEEE Trans. Ind. Appl. 1992, 28, 1345–1352.

[9] Jansen, P.L.; Lorenz, R.D. Transducerless Position and Velocity Estimation in Induction and Salient AC Machines. IEEE Trans. Ind. Appl. 1995, 31, 240–247.

[10] Shinnaka, S. New "D-state-observer"-based vector control for sensorless drive of permanent magnet synchronous motors. IEEE Trans. Ind. Appl. 2005, 41, 825–833.

[11] Piippo, A.; Hinkkanen, M.; Luomi, J. Analysis of an adaptive observer for sensorless control of interior permanent magnet synchronous motors. IEEE Trans. Ind. Electron. 2008, 55, 570–576.

[12] Alahakoon, S.; Fernando, T.; Trinh, H.; Sreeram, V. Unknown input sliding mode functional observers with application to sensorless control of permanent magnet synchronous machines. J. Frankl. Inst. 2013, 350, 107–128.

[13] Kim, H.; Son, J.; Lee, J. A high-speed sliding-mode observer for the sensorless speed control of a PMSM. IEEE Trans. Ind. Electron. 2011, 58, 4069–4077.

[14] Bolognani, S.; Tubiana, L.; Zigliotto, M. Extended Kalman filter tuning in sensorless PMSM drives. IEEE Trans. Ind. Appl. 2003, 39, 1741–1747.

[15] Quang, N.K.; Hieu, N.T.; Ha, Q.P. FPGA-based sensorless PMSM speed control using reduced order extended kalman filters. IEEE Trans. Ind. Electron. 2014, 61, 6574–6582.

[16] Hinkkanen, M.; Tuovinen, T.; Harnefors, L.; Luomi, J. A combined position and stator-resistance observer for salient PMSM drives: Design and stability analysis. IEEE Trans. Power Electron. 2012, 27, 601–609.

[17] Holtz, J. Acquisition of position error and magnet polarity for sensorless control of PM synchronous machines. IEEE Trans. Ind. Appl. 2008, 44, 1172–1180.

[18] DeKock, H.W.; Kamper, M.J.; Kennel, R.M. Anisotropy Comparison of Reluctance and PM Synchronous Machines for Position Sensorless Control Using HF Carrier Injection. IEEE Trans. Power Electron. 2009, 24, 1905–1913.

[19] Jang, J.H.; Sul, S.K.; Ha, J.I.; Ide, K.; Sawamura, M. Sensorless Drive of Surface-Mounted Permanent-Magnet Motor by High-Frequency Signal Injection Based on Magnetic Saliency. IEEE Trans. Ind. Appl. 2003, 39, 1031–1039.

[20] Corley, M.; Lorenz, R. Rotor position and velocity estimation for a salient-pole permanent magnet synchronous machine at standstill and high speeds. IEEE Trans. Ind. Appl. 1998, 34, 784 – 789.

[21] Briz, F.; Degner, M.W.; Garcia, P.; Lorenz, R.D. Comparison of saliency-based sensorless control techniques for AC machines. IEEE Trans. Ind. Appl. 2004, 40, 1107–1115.

[22] Liu, J.M.; Zhu, Z.Q. Sensorless Control Strategy by Square-Waveform High-Frequency Pulsating Signal Injection Into Stationary Reference Frame. IEEE J. Emerg. Sel. Top. Power Electron. 2014, 2, 171–180.

[23] Quanlong Hu; Ling Liu; Cheng Zhang; Lin Cheng. “Researching for Sensorless Control of PMSM Based on a Novel Sliding Mode Observer”. 3rd International Conference on Advanced Robotics and Mechatronics (ICARM), 2018.

[24] Simon Zossak; Marek Stulraiter; Pavol Makys; Martin Sumega. “Initial Position Detection of PMSM”. IEEE 9th International Symposium on Sensorless Control for Electrical Drives (SLED), 2018.

[25] S. Zossak, M. Stulrajter, and P. Makys. “Self-sensing Control of PMSM at Zero and Low Speed”. In: Communications 20.1 (2018), pp. 12–18.

[26] G. Scarletta, G. Scelba, and A. Testa. “High Performance Sensorless Controls Based on HF Excitation: A viable Solution for Future AC Motor Drives?” In: IEEE Workshop on Electrical Machines Design, Control and Diagnosis (WEMDCD). Turkey, 2015.

[27] J. Schuster, V. Ketchedjian, and J. Roth-Stielow. “Sensorless Position Estimation at low Speeds till Standstill with Reduced Filtering Requirements for an Externally Exited Synchronous Machine”. In: Proceedings of International Symposium on Sensorless Control for Electrical Drives. Italy, 2017.

[28] R. Filka, P. Balazovic, and B. Dobrucky. “Transducerless Speed Control with Initial Position Detection for Low Cost PMSM Drives”. In: Proceedings of 13th International Power Electronics and Motion Control Conference (EPE/PEMC 2008). Poland, 2008.

[29] Víctor Repecho ; Domingo Biel ; Antoni Arias. “Enhanced high frequency injection algorithm for sensorless Sliding Mode Control PMSM drives”.2014 IEEE 11th International Multi-Conference on Systems, Signals & Devices (SSD14).

[30] Víctor Repecho ; Domingo Biel ; Antoni Arias. “Fixed Switching Period Discrete-Time Sliding Mode Current Control of a PMSM”. IEEE Transactions on Industrial Electronics (Volume: 65 , Issue: 3 , March 2018).

[31] Vadim Utkin, Juergen Guldner, Jingxin Shi. “Sliding mode control in electro-mechanical systems”. CRC press, 2009, vol. 34.

[32] Zedong Zheng ; Yongdong Li ; Xi Xiao ; Maurice Fadel; “Mechanical sensorless control of SPMSM based on HF signal Injection and Kalman filter”. 2008 International Conference on Electrical Machines and Systems.

[33] Ramakrishnan Rajavenkitasubramony. “Rotor Position and Phase Inductance Estimation of Permanent Magnet Synchronous Machine Using PWM Excitation”. PhD thesis, Automotive Systems Engineering in the University of Michigan-Dearborn 2018.

Glossary

ADC	Analog to Digital Converter
BPF	Band-pass filter
Dq axis	Direct-Quadrature axes
DSP system	Digital Signal Processing System
DTC	Direct Torque Control
FOC	Field Oriented Control
IP-Controller	Integral Proportional Controller
LDTC	Linear Direct Torque Control
LPF	Low-pass filter
MATLAB	Matrix Laboratory
NINFTSMO	Novel Integral Non-singular Fast Terminal Sliding Mode Observer
NSMO	Novel Sliding Mode Observer
PMSM	Permanent Magnet Synchronous Motor
PI-Controller	Proportional Integral Controller
PWM	Pulse Width Modulator
SMC	Sliding Mode Control
SMCC	Sliding Mode Current Control
SMO	Sliding Mode Observer
SVM	Space Vector Modulation